An Ontology Evolution-Based Framework for Semantic Information Retrieval

Miguel Ángel Rodríguez-García, Rafael Valencia-García, and Francisco García-Sánchez

Departamento de Informtica y Sistemas,
Universidad de Murcia, Campus de Espinardo 30100 Murcia, Spain
{miguelangel.rodriguez,valencia,frgarcia}@um.es
http://www.um.es

Abstract. Ontologies evolve continuously during their life cycle to adapt to new requirements and necessities. Ontology-based information retrieval systems use semantic annotations that are also regularly updated to reflect new points of view. In order to provide a general solution and to minimize the users' effort in the ontology enriching process, a methodology for extracting terms and evolve the domain ontology from Wikipedia is proposed in this work. The framework presented here combines an ontology-based information retrieval system with an ontology evolution approach in such a way that it simplifies the tasks of updating concepts and relations in domain ontologies. This framework has been validated in a scenario where ICT-related cloud services matching the user needs are to be found.

1 Introduction

Ontologies constitute the standard knowledge representation mechanism for the Semantic Web [1]. The formal semantics underlying ontology languages enables the automatic processing of the information and allows the use of semantic reasoners to infer new, non-explicit knowledge. In this work, an ontology is seen as "a formal and explicit specification of a shared conceptualization" [2]. Ontologies provide a formal, structured knowledge representation, and have the advantage of being reusable and shareable. They also provide a common vocabulary for a domain and define, with different levels of formality, the meaning of the terms and the relations between them. Ontologies have proven their value in various fields such as information retrieval [3], semantic search [4], service discovery [5] and question answering [6]

Ontology evolution can be defined as the timely adaptation of an ontology to changed business requirements, as well as the consistent management/ propagation of these changes to dependent elements [7]. In fact, the evolution and modification in one part of the domain ontology can produce inconsistencies in the whole ontology [8]. In this context, it is important to distinguish between ontology evolution and ontology versioning. While ontology evolution allows access to all data only through the newest ontology, the ontology versioning allows

P. Herrero et al. (Eds.): OTM 2012 Workshops, LNCS 7567, pp. 163–172, 2012.

access to data through different versions of the ontology [7]. The use of ontology versioning in platforms is gaining success and some platforms such as SHOE [9] and KAON [10] support multiple versions of ontologies and enable to declare whether the new version is backward-compatible with an old version.

In this work, an ontology-based information retrieval system supported by an ontology evolution approach is presented. The proposed framework is capable of simplifying the tasks of updating the concepts and relations that form part of the domain ontologies that comprise the core of the system. The framework has been tested in a scenario where ICT-related cloud services matching the user needs are to be found. The rest of this paper is organized as follows. In Section 2, a detailed description of the whole platform is given. The use case scenario for retrieving services in the cloud is presented in Section 3. Finally, conclusions and future work are put forward in Section 4.

2 Platform Architecture

The architecture of the proposed framework is shown in figure 1. The system is composed of five main modules: (1) the semantic representation and annotation module, (2) the semantic indexing module, (3) the term extractor module, (4) the ontology evolution module and (5) the semantic search engine. In a nutshell the system works as follows. First, natural-language, non-structured texts are

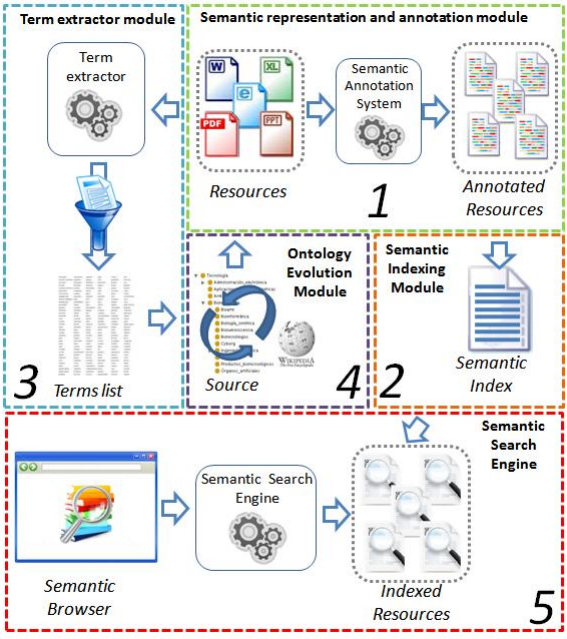

Fig. 1. Platform architecture

semantically represented and annotated by using the domain ontologies. From these annotations a semantic index is then created using the classic vector space model. At the same time, the term extractor obtains the terms appearing in the texts that are not present in the domain ontologies. The ontology evolution module then checks whether the most important new terms previously gathered can be added to the domain ontologies. Finally, a semantic search engine permits to retrieve the matching text from keyword-based searches.

2.1 Semantic Annotation Module (1)

This module receives both the domain ontologies and the natural-language, non-structured texts as inputs. Using a set of natural language processing (NLP) tools, it then obtains a semantic annotation for the analyzed texts in accordance with the domain ontologies. This module is based on the methodology presented in [3] and is composed of two main phases, namely, the NLP phase and the semantic annotation phase.

The main aim of the NLP stage is the extraction of the morphosyntactic structure of each sentence. For this purpose, a set of NLP software tools, including a sentence detection component, a tokenizer, a set of POS taggers, a set of lemmatizers and a set of syntactic parsers, have been developed. Besides, the GATE framework [1] has been employed. GATE is an infrastructure for developing and deploying software components that process human language. GATE helps scientists and developers in three ways: (i) by specifying an architecture, or organizational structure, for language processing software; (ii) by providing a framework, or class library, that implements the architecture and can be used to embed language processing capabilities in diverse applications; (iii) by providing a development environment built on top of the framework made up of convenient graphical tools for developing components. As a result of this first phase, a set of annotations rep-resenting the syntactic structure of the text is obtained.

During the semantic annotation phase texts are annotated with the classes and instances of the domain ontologies by following the process described next. First, the most important linguistic expressions are identified by means of linguistic approaches based on the syntactic structure of the text. For each linguistic expression the system tries to determine whether such expression is an individual or a class of any of the domain ontologies by searching through the hierarchy categories of Wikipedia. If so, the linguistic expression is related with the URI (Uniform Resource Identifier) of the class or instance of such domain ontology.

2.2 Semantic Indexing Module (2)

In this module, the system retrieves all the annotated knowledge from the previous module and tries to create fully-filled annotations with this knowledge. This step is based on the work presented in [11]. Each annotation of each document is stored in a database and is assigned a weight, which reflects how relevant

[1] `http://gate.ac.uk/`

the ontological entity is for the document meaning. Weights are calculated by using the TF-IDF algorithm [12], which satisfies the following equation (see equation 1).

$$(tf - idf)_{i,d} = \frac{n_{i,d}}{\sum_k n_{k,d}} * \log \frac{|D|}{N_i} \quad (1)$$

where $n_{i,d}$ is the number of occurrences of the ontological entity i in the document d, $\sum_k n_{k,d}$ is the sum of the occurrences of all the ontological entities identified in the document d, $|D|$ is the set of all documents and N_i is the number of all documents annotated with i.

In this scenario, the content descriptions comprise the documents to be analysed. For each description, an index is calculated based on the adaptation of the classic vector space model presented in [11]. Each description is represented as a vector in which each dimension corresponds to a separate ontological concept of the domain ontology. The value of each ontological concept dimension is calculated as follows (see equation 2).

$$(v_1, v_2, ..., v_n)_d \quad where \quad v_i = \sum_{j=1}^{n} \frac{tf - idf_j d}{e^{dist(i,j)}} n_{k,d} \quad (2)$$

where *dist(i,j)* is the semantic distance between the concept i and concept j in the domain ontology. This distance is calculated by using the taxonomic (subclass_of) relationships of concepts in the domain ontology. So, the distance between a concept and itself is *0*, the distance between a concepts and its taxonomic parent or child is *1* and so on.

2.3 Term Extractor Module (3)

Through this module, the most significant terms from the text documents are identified. This module is based on previous works of our research group [13]. It is assumed that there exist both multiword and single word terms. By taking into account this assumption, two different methods have been implemented: the NC-Value algorithm [14], which allows to obtain the multiword terms candidates to represent concept, and RIDF [15], which has been employed to obtain terms formed by one word.

The list of the most important terms appearing in the texts that are not part of the domain ontologies, is taken as the input of the ontology evolution module.

2.4 Ontology Evolution Module (4)

The main objective of this module is to maintain and evolve ontologies by using the information available in Wikipedia. The terms list gathered by the term extractor module is used to enrich, enhance and increase the knowledge represented on the domain ontologies. Wikipedia is a free encyclopedia where thousand of concepts are classified in a taxonomy. The main idea of this method is to keep

domain ontologies up-to-date by exploiting the structure of Wikipedia, that is, the assignment of articles to categories, the subcategory relation and the cross-language relations linking equivalent articles across languages. The Wikipedia structure is an important and necessary condition since it allows to establish group of terms defining the semantic concepts. This organization is not yet available in DBPedia for all languages. Each relevant term in the list produced by the term extractor module (i.e. terms not currently present in the ontologies) is looked up in Wikipedia. If a Wikipedia article or category is found matching the term, a new concept is created containing all the term's synonyms in both English and Spanish and subsequently added into the domain ontology by using the algorithm described in figure 2.

This algorithm has been implemented by using the OWL API [2] framework and Java Wikipedia API (Bliki engine [3]), which is a parser library for converting Wikipedia wikitext notation to HTML. The module has been designed to allow the access to online wikitext using the Hypertext Transfer Protocol (HTTP), and so the extracted content is always up-to-date since it is gathered from the online version of Wikipedia. Additionally, the tool permits to collect outdated wikitext content if desired.

Figure 2 shows the pseudocode of the algorithm designed to solve the ontology evolution problem. The input of the algorithm is the list of terms produced by the term extractor module. Each term in the list represents a candidate concept that can be used to enrich the ontology knowledge.

The proposed evolution algorithm has been designed to find a path in the Wikipedia hierarchy. This path assists the system to generate relationships between the candidate concepts and the elements already available in the ontology. The process of this algorithm comprises two important phases. In the first phase the system is responsible for extracting all the information about all the classes in the ontology. This information is thus a compact representation of all the concepts in the domain. During the second phase the algorithm attempts to find the shortest path between each term in the list and the ontology concepts based on the Wikipedia categories. An ontology class is defined during the search process for each category that is obtained from the Wikipedia hierarchy. These classes are created by extracting the synonymous terms from Wikipedia, which are used to define the concept. In fact, the algorithm searches for Wikipedia categories that are shared by an ontology concept and one of the relevant terms in the list.

2.5 Semantic Search Engine Module (5)

This module is in charge of finding semantic concepts in different kinds of textual resources from a keyword-based query. This process takes advantage of the semantic content and annotations previously generated by the system.

First, users introduce a number of keywords. Next, the system identifies what concepts in the domain ontology are referred to by those keywords. Each

[2] `http://owlapi.sourceforge.net/`
[3] `http://code.google.com/p/gwtwiki/`

```
Declare a list of ontology nodes nodesOntology
Obtain all the ontology concepts (classes)from the 'ontology' resource
Declare a list of terms termList
Obtain the list of terms from term extractor module
FOR each term DO
   the next term is obtained
   IF the next term is not already in the ontology THEN
      FOR each node DO
         the node concept is obtained
         the first concept is built from Wikipedia
         IF first concept is null THEN
            //If there is not any concept in the Wikipedia then
            //we have to continue with the next node
            continue;
         ELSE
            the second concept is built from Wikipedia.
            IF second concept is null THEN
               //If there is not any term in the Wikipedia then
               //we have to continue to the next term
               break;
            ELSE
               joining concepts from Wikipedia
            END IF
         END IF
      END DO
   END IF
END DO
```

Fig. 2. Algorithm for the ontology evolution process

document is represented as a vector in which each dimension corresponds to a separate concept of the domain ontology. The semantic search engine then calculates a similarity value between the query q and each semantic concept. In order to do that, the cosine similarity is used (see equation 3).

$$sim(q, s) = \cos \vartheta = \frac{q \bullet s}{|q| \bullet |s|} \tag{3}$$

A ranking with the most relevant semantic concepts that are related to the topics referenced in the query is then defined by using the similarity function showed in equation 3. In figure 3, the vector space model calculation is shown graphically. The first vector (s) is the vector calculated by equation 2 for each content description and the second vector (q) is the one created from the concepts extracted from search engine query. The ϑ symbol is the angle between both vectors, which represents the similarity degree between two documents.

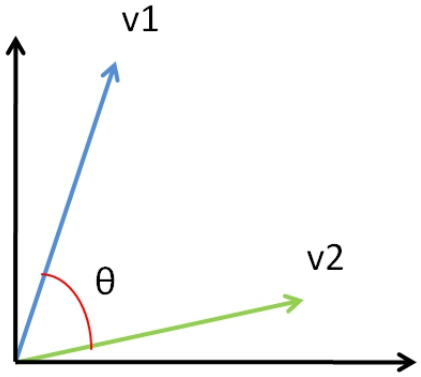

Fig. 3. Platform architecture

3 Use Case Escenario: Retrieving Services in the Cloud

The platform described in the previous section has been implemented and later tested in an ICT domain-based cloud services retrieval use case scenario. For this, in the first place, around 300 different ICT-related cloud services with their description in natural language have been selected to be automatically annotated by the system.

Initially, the ICT ontology was empty. So, no semantic annotation can be obtained during this first stage. The term extractor module is then executed and a list of terms containing the most relevant terms that appear in the cloud services descriptions is generated. Next, the domain ontology is enriched by the ontology evolution module, which takes into account the concepts in Wikipedia that corresponds to each term in the list. As a result, an ontology formed by 10 classes and 40 *"subclass-of"* relationships is produced. In figure 4, an excerpt of the ontology is shown.

Once the initial domain ontology is available, the services descriptions can be semantically annotated and stored in the ontology repository. The Virtuoso repository has been used to implement the ontology repository. A semantic index is then calculated for each service by means of the semantic indexing module. Once the semantic indexes have been created, the system is ready to be queried. The ultimate goal of this experimental evaluation is to elucidate whether the semantic search engine module of the proposed platform is useful. Five topic-based queries were issued. For each query, a set of cloud services was manually selected. At the same time, the semantic search engine was asked to perform the same task, in an automatic way. The outcome of the semantic search engine was then compared against those produced by the manual selection.

Fig. 4. Platform architecture

The results are shown in Table 1. The system obtained the best results for queries about the topic "Databases", with a precision of 0.85, a recall of 0.77 and a F1 measure of 0.81. In general, the system obtains better results in precision (80% on average) than in recall (76% on average). Anyhow, these results of the proposed system seem very promising.

Table 1. Precision, recall and F1 of the experiment

Topics	Precision	Recall	F1
Databases	0,85	0,77	0,81
Java	0,81	0,76	0,78
Storage	0,75	0,79	0,77
Backup	0,8	0,73	0,76

4 Conclusions and Discussion

Semantic-based information retrieval differs from traditional information retrieval systems because informational resources in the former approach are given formal meaning. Certainly, the added value of semantic information retrieval with respect to traditional keyword-based retrieval relies on the use of ontologies to make explicit the semantics of the resources that are being searched for. However, this novel and powerful approach could be hampered by the changes in the domain in which the system is being used. Therefore, it is of utmost importance to continuously update the ontologies to satisfactorily reflect these changes in the domain. Ontology evolution is the research field concerned with the timely adaptation of ontologies to changing requirements.

In this work, we propose an ontology evolution-based framework for semantic information retrieval. The global resulting platform combines an ontology evolution approach with a semantic-based information retrieval system. This integrated approach allows to automatically keep ontologies up-to-date so that the

retrieval tool is constantly accurate during its functioning. In order to evaluate the usefulness of the proposed platform, it has been tested in a real use case scenario. The testing environment consists of the search for ICT-related cloud services from keyword-based queries. The results of the experiments described above show that the value of the system presented here is twofold: (1) it drastically reduces the time-consuming effort of manually extending the ontologies with new concepts and relations to adapt to changing requirements, and (2) it accurately retrieves the services that match a user' query. A further benefit of the approach presented here is the use of Wikipedia as source of information. The structure, organization, dynamism and trustability of Wikipedia's contents enable the system to develop a comprehensive and up-to-date ontology.

The main limitation of the proposed framework is that the "semantic representation and annotation module" only works with unstructured, natural-language text. We plan to extend this module so as to include support for more structured content (e.g. database tables). Additionally, we are currently working on a more ambitious experiment that covers all kinds of cloud services regardless of their application domain. For future work, we aim to improve the approach in a number of ways. First, we intend to broaden the scope of the approach to other application domains such as e-health and e-learning. The ontology evolution process that is supported by our framework can help gathering more up-to-date knowledge to be applied in medical diagnosis or in an e-learning environment. A further enhancement of the framework that is being considered is the use of other information repositories besides Wikipedia. It cannot be expected that all the searched concepts are in Wikipedia. It could be appropriate to consider other trustworthy information sources when a particular concept is not found in Wikipedia.

Acknowledgments. This work has been supported by the Spanish Ministry for Science and Innovation under project SeCloud: TIN2010-18650.

References

1. Shadbolt, N., Berners-Lee, T., Hall, W.: The semantic web revisited. IEEE Intelligent Systems 21(3), 96–101 (2006)
2. Studer, R., Benjamins, V.R., Fensel, D.: Knowledge engineering: Principles and methods. Data Knowledge Engineering 25(1-2), 161–197 (1998)
3. Valencia-García, R., Fernández-Breis, J.T., Ruiz-Martínez, J.M., García-Sánchez, F., Martínez-Bjar, R.: A knowledge acquisition methodology to ontology construction for information retrieval from medical documents. Expert Systems 25(3), 314–334 (2008)
4. Lupiani-Ruiz, E., García-Manotas, I., Valencia-García, R., García-Sánchez, F., Castellanos-Nieves, D., Fernández-Breis, J.T., Camón-Herrero, J.B.: Financial news semantic search engine. Expert Systems with Applications 38(12), 15565–15572 (2011)
5. García-Sánchez, F., Valencia-García, R., Martínez-Béjar, R., Fernández-Breis, J.T.: An ontology, intelligent agent-based framework for the provision of semantic web services. Expert Systems with Applications 36(2), 3167–3187 (2009)

6. Valencia-García, R., García-Sánchez, F., Castellanos-Nieves, D., Fernández-Breis, J.T.: Owlpath: An owl ontology-guided query editor. IEEE Transactions on Systems, Man, and Cybernetics, Part A 41(1), 121–136 (2011)
7. Maedche, A., Motik, B., Stojanovic, L., Stojanovic, N.: User-driven ontology evolution management, pp. 285–300. Springer (2002)
8. Haase, P., Haase, P., Stojanovic, L.: Consistent evolution of owl ontologies, pp. 182–197. Springer (2005)
9. Heflin, J., Hendler, J.A.: Dynamic ontologies on the web. In: Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence, pp. 443–449. AAAI Press (2000)
10. Bozsak, E., Ehrig, M., Handschuh, S., Hotho, A., Maedche, A., Motik, B., Oberle, D., Schmitz, C., Staab, S., Stojanovic, L., Stojanovic, N., Studer, R., Stumme, G., Sure, Y., Tane, J., Volz, R., Zacharias, V.: KAON - Towards a Large Scale Semantic Web. In: Bauknecht, K., Tjoa, A.M., Quirchmayr, G. (eds.) EC-Web 2002. LNCS, vol. 2455, pp. 304–313. Springer, Heidelberg (2002)
11. Castells, P., Fernández, M., Vallet, D.: An adaptation of the vector-space model for ontology-based information retrieval. IEEE Trans. on Knowl. and Data Eng. 19(2), 261–272 (2007)
12. Salton, G., McGill, M.J.: Introduction to Modern Information Retrieval. McGraw-Hill, Inc., New York (1986)
13. Ochoa, J.L., Ángela Almela, M.L.H.A., Valencia-García, R.: Learning morphosyntactic patterns for multiword term extraction. Scientific Research and Essays 6(26), 5563–5578 (2011)
14. Frantzi, K.T., Ananiadou, S., Tsujii, J.: The $C - value/NC - value$ Method of Automatic Recognition for Multi-word Terms. In: Nikolaou, C., Stephanidis, C. (eds.) ECDL 1998. LNCS, vol. 1513, pp. 585–604. Springer, Heidelberg (1998)
15. Church, K., Gale, W.A.: Inverse document frequency (idf): A measure of deviations from poisson. In: Proceedings of the Third Workshop on Very Large Corpora, pp. 121–130 (1995)