Knowledge Source Discovery: An experience using Ontologies, WordNet and Artificial Neural Networks

M. Rubiolo[1], M.L. Caliusco[2], G. Stegmayer[2], M. Gareli[1] and M. Coronel[1]

[1] CIDISI-UTN-FRSF, Lavaise 610, Santa Fe, Argentina
[2] CONICET, CIDISI-UTN-FRSF, Lavaise 610, Santa Fe, Argentina

Abstract. This paper describes our continuing research on ontology-based knowledge source discovery on the Semantic Web. The research documented here is focused on discovering distributed knowledge sources from a user query using an Artificial Neural Network model. An experience using the Wordnet multilingual database for the translation of the terms extracted from the user query and for their codification is presented here. Preliminary results provide us with the conviction that combining ANN with WordNet has clearly made the system much more efficient.

1 Introduction

The web grows and evolves at a fast speed, imposing scalability problems to web search engines [1]. Moreover, another ingredient has been recently added: data semantics represented by means of ontologies [2]. Ontologies have shown to be suitable for facilitating knowledge sharing and reuse. Thus, the new *Semantic Web* allows searching not only information but also knowledge. The knowledge source discovery task in such an open distributed system presents a new challenge due to the lack of an integrated view of all the available knowledge sources [3].

The web of the future will consist of small highly contextualized ontologies developed with different languages and different granularity levels [4]. The distributed development of domain-specific ontologies introduces another problem: in the Semantic Web many independently developed ontologies co-exist describing the same or very similar fields of knowledge. This can be caused, among other things, by the use of different natural languages (Paper vs. Artículo), different technical sublanguages (Paper vs. Memo), or the use of synonyms (Paper vs. Article). That is why, ontology-matching techniques are needed, that is to say, semantic affinity must be identified between concepts belonging to different ontologies [2].

A matching problem can be viewed as a classification problem and an ANN-based classifier can be used for this task [5]. The model should classify a term as a word belonging (or not) to a domain. To achieve this, the data need to undergo an adequate pre-processing step before entering the neural model [6]. In this work, we propose an ANN-based ontology-matching model, and the use of WordNet for codifying terms as an appropriate domain data representation within the ANN-based model.

The main contribution to the field of this paper is to share with the community the results of an experience in: a) using WordNet multilingual corpus to codify the domain data, which is useful for improving a traditional web search by considering (indirectly) terms synonyms and translation into different languages; and b) using this appropriate codified data to achieve the benefits of the application of an ANN-based ontology-matching model.

The paper is organized as follows. In section 2, the knowledge source discovery task is explained. Section 3 presents the proposed ANN-based ontology-matching model in detail. The results of the model evaluation and comparison against an ontology-matching algorithm called H-Match as well as a discussion of the experiments are shown in Section 4. Finally, section 5 presents the conclusions of this contribution.

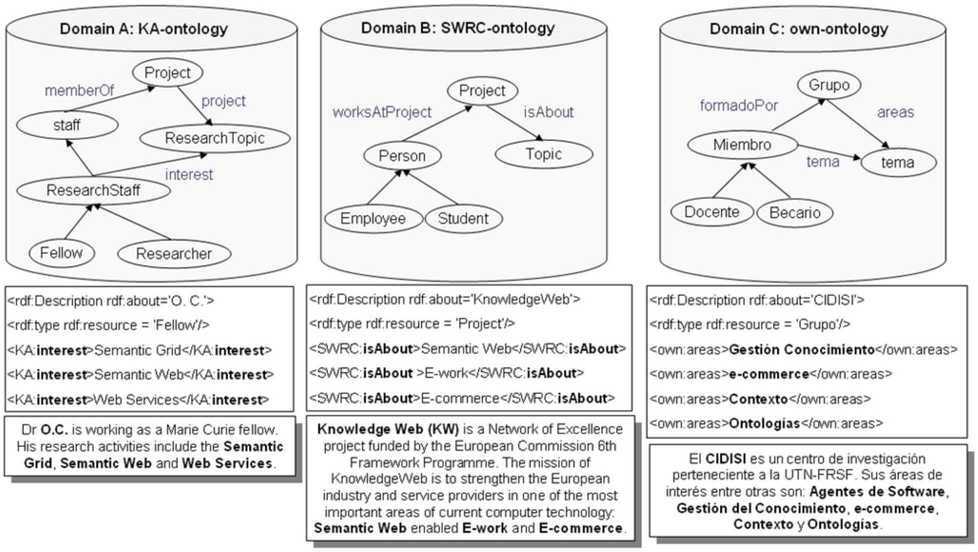
2 Knowledge Source Discovery on the Semantic Web

In open distributed systems such as the Semantic Web several nodes (domains) need resources and information (i.e. data, documents, services) provided by other domains in the net. Such systems can be viewed as a network of several independent nodes having different roles and capacities. In this scenario, a key problem is the dynamic discovery of knowledge sources (i.e. the capacity of finding knowledge sources in the system about resources and information) that, in a given moment, respond well to the requirements of a node request [3].

Searching on the Semantic Web differs in several aspects from a traditional web search, especially because of the structure of an online collection of documents. Traditional search machines do not try to understand the semantics of the indexed documents. Conventional retrieval models, based on the matching of terms between documents and the user queries, often suffer from either missing relevant documents not indexed by the keywords used in a query but by synonyms, or retrieving irrelevant documents indexed by unintended sense of the keywords in the query. Instead, retrieval models should not only find the right information in a precise way, but also find or infer related knowledge.

2.1 Motivating Scenario

In [7], an architecture for discovering knowledge sources on the Semantic Web was proposed, composed by mobile agents, the Knowledge Source Discovery (KSD) agent and the domains. The mobile agents receive the request from the user and look for an answer visiting the domains according to a list generated by the KSD. The KSD agent has the responsability for knowing which domains can provide knowledge inside a specific area, and it indicates a route to mobile agents that carry a user request. The KSD agent knows the location (url) of the domains that can provide knowledge, but it does not provide the knowledge nor the analysis of what the domain contains (files, pictures, documents, etc.).



```
<rdf:Description rdf:about='O. C.'>
<rdf:type rdf:resource = 'Fellow'/>
<KA:interest>Semantic Grid</KA:interest>
<KA:interest>Semantic Web</KA:interest>
<KA:interest>Web Services</KA:interest>
```

Dr **O.C.** is working as a Marie Curie fellow. His research activities include the **Semantic Grid**, **Semantic Web** and **Web Services**.

```
<rdf:Description rdf:about='KnowledgeWeb'>
<rdf:type rdf:resource = 'Project'/>
<SWRC:isAbout>Semantic Web</SWRC:isAbout>
<SWRC:isAbout >E-work</SWRC:isAbout>
<SWRC:isAbout>E-commerce</SWRC:isAbout>
```

Knowledge Web (KW) is a Network of Excellence project funded by the European Commission 6th Framework Programme. The mission of KnowledgeWeb is to strengthen the European industry and service providers in one of the most important areas of current computer technology: **Semantic Web** enabled **E-work** and **E-commerce**.

```
<rdf:Description rdf:about='CIDISI'>
<rdf:type rdf:resource = 'Grupo'/>
<own:areas>Gestión Conocimiento</own:areas>
<own:areas>e-commerce</own:areas>
<own:areas>Contexto</own:areas>
<own:areas>Ontologías</own:areas>
```

El **CIDISI** es un centro de investigación perteneciente a la UTN-FRSF. Sus áreas de interés entre otras son: **Agentes de Software**, **Gestión del Conocimiento**, **e-commerce**, **Contexto** y **Ontologías**.

Fig. 1. Domains belonging to the R+D field and their semantic annotations.

The other components of the architecture are the domains. Each domain has its own ontology used to semantically markup the information published in their websites. Suppose there are three domains (A, B, and C) which belong to the Research & Development (R+D) field of knowledge (figure 1). The domain A uses the KA-ontology [3]. The domain B uses the Semantic Web for Research Communities (SWRC) ontology [4]. Finally, the domain C uses an own highly-specialized model. As can be seen, each domain may use a different ontology to semantically annotate the provided information even if they belong to the same field of knowledge.

Resource Description Framework (RDF) is used to define an ontology-based semantic markup for the domain website. Each RDF-triplet assigns entities and relations in the text linked to their semantic descriptions in an ontology. For example, in the domain A, the following RDF-triplets: <`O.C., interest, Semantic Grid`>, <`O.C., interest, Semantic Web`> and <`O.C., interest, Web Services`> represent the research interests of O.C. described in the text (see the left area of figure 1).

The KSD agent must be capable of dynamically identifying which domains could satisfy a request brought to it by a mobile agent. This dynamic knowledge discovery requires models and techniques which allow finding ontology concepts that have semantic affinity among them, even when they are syntactically different. In order to do this, the KSD agent has to be able to match (probably different) domain ontologies. To face this ontology-matching problem, we propose the use of an ANN model with

[3] http://protege.cim3.net/file/pub/ontologies/ka/ka.owl
[4] http://ontoware.org/projects/swrc/

supervised learning stored in the KSD agent Knowledge Base (KB) and trained (and re-trained periodically) off-line.

ANNs are information processing systems inspired by the ability of the human brain to learn from observations and to generalize by abstraction. Knowledge is acquired by the network through a learning process, and the connection strengths between neurons, known as synaptic weights, are used to store this knowledge. Training a neural network means adapting its connections so that the model exhibits the desired computational behavior for all input patterns. Selection of training data plays a vital role in the performance of a supervised ANN. Generally, rather than focusing on volume, it is better to concentrate on the quality and representational nature of the data set. A good training set should contain routine, unusual and boundary-condition cases [8].

The KSD agent must also be capable of understanding the natural-language-based query received from the client, which is translated into an RDF-triplet (this process is out of the scope of this work). The resultant RDF-triplet is codified before entering the ANN-based matching model proposed in this work. A WordNet Corpus, which could be composed of different-languages WordNet databases, is used by the KSD agent for this task.

WordNet is a lexical database for the English language [9]. It groups English words into sets of synonyms called *synsets*. Every synset contains a group of synonymous words or collocations (sequence of words that together form a specific meaning); different senses of a word are in different synsets. The meaning of the synsets is further clarified by short defining glosses. Most synsets are connected to other synsets via a number of semantic relations that vary according to the type of word, and include synonyms, among others. This research uses WordNet[5] 1.6 since different wordnets for several languages (such as Spanish[6]) are structured in the same way.

3 Ontology-matching: ANN-based model and training

This section presents, through an example, the proposed neural network model for ontology matching and its training strategy,

3.1 ANN-based model

For neural networks, a matching problem can be viewed as a classification problem. Our ANN-based matcher uses schema-level information and instance-level information (RDF-triplet instances belonging to the RDF annotations of the ontology domain) inside the X ontology to learn a classifier for domain X, and then it uses schema-level information and instance-level information inside the Y ontology to learn a classifier for domain Y. It then classifies instances of Y according to the X classifier, and vice-versa. Hence, we have a method for identifying instances of X $\bigcap$ Y. The same idea is applied to more than two domains.

For building a classifier for Domain X, its RDF-triplets are extracted. Each part of the triplet corresponds to an input unit for the neural model. This way, the proposed model has 3 inputs, each input corresponding to each triplet component. The proposed model is a multilayer perceptron (MLP) neural network model. The outputs of the model are as much neurons as domains. For example, having domain X, Y and Z,

[5] http://wordnet.princeton.edu/

[6] http://www.lsi.upc.edu/ nlp

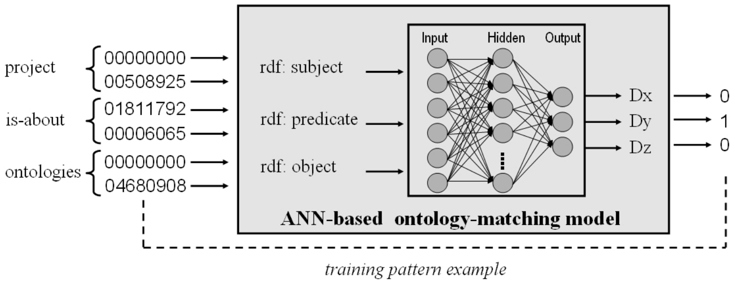


Fig. 2. ANN-based ontology-matching model and its training patterns example.

the ANN-model has 3 output neurons. The first neuron will be activated each time a RDF-triplet belonging to the domain X is presented to the model. The second neuron will be activated each time a RDF-triplet belonging to the domain Y is presented to the model, and so forth. The activation of a neuron consists in producing a value of (near) 1 when the input RDF-triplet exists in the corresponding ontology domain, and 0 otherwise.

3.2 Training data

For training the ANN-based ontology matching model, training examples must be formed. Each example must tell the network that a certain RDF-triplet can be found in a certain domain. This is done through training patterns, which must be numbers. Once the RDF-triplets are identified for each domain, they have to be codified from string to numbers.

We propose a way of codifying the terms using the WordNet database, where a term is associated with a code named *synset offset*. This code is represented by an 8 digit decimal integer. In this way, an appropriate pattern-codification schema can be achieved because all terms can be codified with an invariant-length code.

However, most of the terms represented in WordNet are single words, not collocations. For example, in English WordNet 1.6 the term *Semantic Web* is not a collocation. This is a problem for term codification that is addressed assuming the collocation as two independent words. Then, a triplet term can be represented as a pair of codes, whose values will vary if the term has a) *a single word:* the term code will be formed by the *synset* code associated with the word, including an 8-zero-code in the first position, representing the absence of another word; b) *two words:* the term code will be formed by the composition of a *synset* code associated with each word.

The proposed model uses the standard backpropagation algorithm for supervised learning, which needs {input/output target} pairs named *training patterns* [10]. They are formed by showing to the model, during training, an input pattern of the form: `InputPattern = <rdf:subject; rdf:predicate; rdf:object>`, with its corresponding target value, indicating to which domain is belongs: `OutputPattern = <Dx; Dy; Dz>` (see Figure 2).

The training data are normalized into the activation function domain of the hidden neurons, before entering the model, since this significantly improves training time and model accuracy.

3.3 Training example

A simple example of one training pattern is presented in figure 2. Considering the ontologies of figure 1, a training pattern indicating that the triplet <*project; is-about; ontologies*> can be found on the Domain B ontology but not on A or C is:
`InputPattern=<project;is-about;ontologies>` and `TargetPattern=<0;1;0>`.

This means that, given the fact that there are projects in the domain B whose research interest is about ontologies, its corresponding triplet would be `<project; is-about; ontologies>` and its corresponding output target would be `<0; 1; 0>`: only the second vector value (that represents Domain B) is equal to 1, indicating that this triplet can be found on domain B ontology.

The Figure 2 shows also the triplet codification. The code related to *is-about* is formed by *is* code `01811792` and *about* code `00006065`: <`01811792;00006065`>. For *project* the code is formed as a combination of zero and `00508925`: <`00000000;00508925`>. In summary, this training pattern would be: `InputPattern = <<00000000;00508925>; <02579744;00006065>; <00000000;04680908>>` and `TargetPattern = <0;1;0>`.

An interesting fact related to the use of different language ontologies arises as a consequence of using the WordNet database for triplet term codification. Because all of the words and their translations are codified with the same code in the WordNet database, the process of identifying the right domain for a particular triplet can be significantly improved. There is some sort of automatic triplet expansion and translation as a consequence of using the WordNet codification scheme for ANN model training. That is to say, a term in a triplet has the same code in English WordNet as well as in Spanish WordNet. For example, the English term *project* and its Spanish translation *proyecto* has the same code: `00508925`. Using this unique code, it is possible to consult all domains, without taking each domain language into account. Similar conclusions can be drawn in the case of synonyms.

4 Evaluating the proposed strategy against H-Match

One very important aspect of evaluation is the data set used for performing it. Datasets for matching ontologies are not easy to find. The first problem is that they require public and well-designed ontologies with meaningful overlap. The data sets made for OAEI [7]. (Ontology Alignment Evaluation Initiative) campaign can be considered as correct by construction but they are not realistic nor very hard. In addition, all data sets defined for evaluating matching algorithm are composed by a pair of ontologies. In contrast, to evaluate the proposed method more than two ontologies are required. *Nano: Falta referencia a Pavel*)

Results of an experience using the ontologies shown in Section 2.1, the WordNet use for codifying triplet terms from the original queries, and the ANN-based ontology-matching model application are reported in this section.

The MLP model parameters are set according to typical values, randomly initialized. The number of input neurons for the MLP model is set to 6, considering a double-code for each triplet term. The hidden layer neuron number is set empirically, according to the training data and the desired accuracy for the matching. At the output, there is a specialized output neuron in the model for each domain. The allowed values for each output neuron are 1 or 0, meaning that the neuron recognizes or not a concept belonging to the domain it represents.

[7] http://oaei.ontologymatching.org/

The ANN-based ontology-matching model proposed is trained with each domain ontology RDF-annotations and their corresponding instances. Since we need a populated ontology, we have semantic annotated three different web pages obtaining 134 patterns for the ANN model training. The ANN model is built and trained off-line, and its parameters are tuned and re-trained periodically when data changes or new domains are added to the system.

Is difficult to make a large results comparison of our proposal against others matching algorithm due to matching algorithms works on structured ontologies. That is to say, as we need a populated ontologies to train our model, we can not apply the most of the matching algorithms because they are focus on structured ontologies.

The proposed ANN-based ontology-matching model has been compared with the H-Match [11], an algorithm for matching populated ontologies by evaluating the semantic affinity between two concepts considering both their linguistic and contextual affinity.

4.1 H-Match algorithm

In order to use this algorithm, a probe query (one word) is sent to each domain, which applies the algorithm to determine whether it has concepts matching it or not. The six examples are later evaluated domain by domain setting the algorithm parameters as: matching model = *intensive*, mapping = *one-to-one*, adopts inheritance = *false*, empty context strategy = *pessimistic*, matching strategy = *standard (asymmetric)* and weight linguistic affinity = *1.0*. The H-Match algorithm provides a semantic affinity value ($S_{i,D}$) for each triplet-term i compared with each domain ontology D. These values are combined to obtain an average matching measurement (Av) for each complete triplet $t_{i,j,k}$ against a domain ontology, according to $Av_{(t_{i,j,k},D)} = \frac{S_{i,D}+S_{j,D}+S_{k,D}}{3}$. To determine if the triplet can be indicated as belonging or related to the analyzed domain D, the semantic affinity measurement $Av_{(t_{i,j,k},D)}$, as well as two of the semantic affinity values, have to be higher than an empirically set threshold of 0.7. If both conditions are satisfied, the triplet $t_{i,j,k}$ is considered to be ”matched” to the domain ontology D.

4.2 Comparison results

The results of the analysis against H-Match algorithm are reported in Table 1. The first column indicates the triplet query considered in the test and the second column indicates which domain it should be associated with.

Table 1. Ontology-matching results comparison

$Query$	$Domain$	$ANN - model$	$H - Match$
1) <fellow,interest,semanticWeb>	A	A	A
2) <miembro,tema,gobierno>	C	C	A,C
3) <project,is-about,ontologies>	A,B	A,B	A,B
4) <researcher,topic,web>	B	C	A,B,C
5) <-,-,semanticGrid>	A	A	A,B,C

The third column reports the results obtained from the use of the proposed ANN-model for the ontology-matching task, while the fourth column reports the results from the use of the H-Match algorithm.

From the results shown in Table 1, it can be stated that the proposed model can be quite accurate for indicating potential domains that can answer a query, compared to a traditional matching algorithm.

Note that the query triplet 3) <*project,is-about,ontologies*> has a translation in both domain A and domain B ontologies, and in fact the ANN model indicates that the domain ontologies of A and B contain some ontology labels or instances that are similar to the presented request.

Another interesting test queries are 2)<*miembro,tema,gobierno*> and 4)<*researcher, topic,web*>. As can be noted, the two first triplets components are translations of the same words and the ANN-based model provides the same answer for both cases, showing the advantage of using a codification scheme for words which is independent of the language. However, here the neural model shows a flaw: the domain C is indicated as the final result because it is the last domain examples the ANN model has seen during the training process. It is an indication that some procedure must be used during training for assuring model independency of trainingg patterns order, such as bootstrapping, cross-validation or leave-one-out algorithms. For all the remaining tests, the ANN-based ontology-matching model has provided satisfactory results.

5 Conclusions

The ontology-based knowledge source discovery on the Semantic Web, focused on discovering distributed knowledge sources from a user query using ANN models and WordNet multilingual database, was experienced in this paper. Using the Wordnet multilingual database for codifying the triplet terms, extracted from the user query, some sort of automatic triplet expansion and translation arose, which improved the traditional search task. This codification also allowed appropriately representing the ANN-based ontology-matching model input data. This paper has shown the benefits of including an ANN-based ontology-matching model inside a KSD agent, whose capabilities for discovering distributed knowledge sources have been improved. In addition, the combination of ANN with WordNet has clearly made the system much more efficient.

References

1. Baeza-Yates, R.: Web mining. Proceedings of LA-WEB Congress **1**(2) (2005) 19–22
2. J. Davies, R. Studer, P.W.: Semantic Web Technologies: trends and research in ontology-based systems. (2007)
3. Castano, S., Ferrara, A., Montanelli, S.: Dynamic Knowledge Discovery in Open, Distributed and Multi-Ontology Systems: Techniques and Applications. (2006)
4. Hendler, J.: Agents and the semantic web. IEEE Intelligent Systems **16(2)** (May 2001) 30–37
5. Caliusco, M., Stegmayer, G. In: Semantic Web Technologies and Artificial Neural Networks for Intelligent Web Knowledge Source Discovery. Springer Verlag (2009) In press

6. Mandl, T.: Implementation of large backpropagation networks for text retrieval. Proceedings of the 3rd International DataAnalysis Symposium (1999) 19–22
7. Stegmayer, G., Caliusco, M., Chiotti, O., Galli, M.: ANN-agent for Distributed Knowledge Source Discovery. On the Move to Meaningful Internet Systems. (2007)
8. Haykin, S.: Neural Networks: A Comprehensive Foundation. (1999)
9. Fellbaum, C.: WordNet: An Electronic Lexical Database (Language, Speech, and Communication). The MIT Press (May 1998)
10. J.Wray, G.Green: Neural networks, approximation theory and precision computation. Neural Networks **8**(1) (1995) 31–37
11. Castano, S., Ferrara, A., Montanelli, S., Racca, G.: Semantic information interoperability in open networked systems. Semantics for Grid Databases, First International IFIP Conference, ICSNW 2004, Paris, France, June 17-19, 2004, Revised Selected Papers (2004) 215–230