ANN-agent for distributed Knowledge Source Discovery

G.Stegmayer,[1] M.Caliusco,[1] O.Chiotti[1][2] and M.Galli[1][2]

[1] CIDISI-CONICET, Lavaise 610, 3000, Santa Fe, Argentina
[2] INGAR-CONICET, Avellaneda 3654, 3000, Santa Fe, Argentina

Abstract. This paper presents an Artificial Neural Network-based (ANN) agent with capabilities for discovering distributed knowledge sources. This agent has the responsibility of guiding knowledge requirements towards distributed ontology domains in the Semantic Web, indicating i.e. potentially partners for the establishment of a dynamic collaborative relationship. The agent performs its task through a neural network model. This work shows a proposed reference architecture, explains the ontology-matching model ANN-based for the agent and presents preliminary good results obtained.

1 Introduction

The Web grows and evolves faster than one would like and expect, imposing scalability and relevance problems to Web search engines [1]. Recently, another ingredient is being added to the Web as well: data semantics. The new Web allows not only searching for information but also knowledge, as well as the establishment of relationships between potential partners for cooperation. The purpose of the Semantic Web is to introduce structure and semantic content in the huge amount of unstructured or semi-structured information available on the Web, being the central notion behind the Semantic Web that of *ontologies*, which describe concepts and their relations in a particular field of knowledge [2].

The knowledge source discovery task in such an open distributed system is a new challenge, because of the lack of an integrated view of all the available sources. When a part of the system wants to initiate, for example, a dynamic collaborative relationship with another one, it is difficult to know who to contact and where to go and look for the required knowledge. Besides that, the distributed development of domain-specific ontologies introduces another problem: in the Semantic Web many independently developed ontologies, describing the same or very similar fields of knowledge, co-exist. Those ontologies are not identical or present minor differences, such as different naming conventions, or higher level differences in their structure and in the way they represent knowledge. Therefore, ontology-matching techniques are needed, that is to say, semantic affinity must be identified between concepts that, appearing in different ontologies, are related.

James Hendler, one of the Semantic Web pioneers, has predicted that in the future each website, organization or business will have its own domain ontology. The Web of the future will be composed by small highly contextualized ontologies, developed with different languages and different granularity levels. The simpler document in the Semantic Web will be composed by concrete facts, classes, properties definitions and metadata [11]. The websites will have, besides an ontology domain to describe the knowledge they can provide, an adequate structure to receive mobile agents (i.e. an agent server) that will travel the net, for example, looking for knowledge required by an end-user. Considering that these agents will have their own ontology for communication, in this future context, the main challenge will be how to go to the most relevant node (avoiding waste of time) and how to communicate with the node.

In this scenario, appears the need for an agent, capable of dynamically identifying the location of distributed knowledge sources, relevant to a request. This agent should have an important routing capability, it should route agents that look for resources to the corresponding site. In addition, this agent should have a way for defining ontology-matching, which is a big problem the agents have to deal with in the Semantic Web. In this work, we propose such an agent and we named it Knowledge Source Discovery (KSD) agent. Our assumption is that, to perform its tasks efficiently, the agent should be specialized in a specific domain context (in this paper, research and development R+D), where we assume that different ontologies will manage similar concepts and content. We think that this fact, combined with supervised learning by example (i.e. Artificial Neural Networks), can be better used by the agent to provide a more effective (mobile agent responsible for searching the web wouldn't loose time, they would visit only domains that could provide an answer) and more efficient (response time would diminish and the knowledge would be available just-in-time for the system users) response, compared to traditional information retrieval mechanisms.

The structure of this paper is the following: the next section introduces some issues regarding searching on the Semantic Web, including a reference architecture and related works. Section 3 explains in detail de agent proposed in this work. In section 4, details on the neural model used by the agent are shown. Finally, section 5 presents some discussions and future work.

2 Searching the Semantic Web

In open distributed systems, several nodes (domains), probably distributed among different organizations, need resources and information (i.e. data, documents, services) provided by other domains in the net. Therefore, an open distributed system can be defined as networks of several independent nodes, having different roles and capacities. In this scenario, a key problem is the dynamic discovery of knowledge sources, understood as the capacity of finding knowledge sources in the system about resources and information, that, in a given moment, better response the requirements of a node request [4]. For example, a researcher from a university wanting to collaborate on the *Semantic Web* topic with another researcher in another university, that is to say, a dynamic collaborative relationship that could be established in a specific context (Research and Development - R+D). The most common example are the organizations or industries in the Web, where there are outsourced services or distributed subsystems (domains) which, during a determined period of time, must collaborate to reach a common goal. Each domain exposes to the system the information it wants to share (through a web-site) together with the corresponding meta-data (the domain ontology) and interacts with other domains trying to acquire/provide resources and knowledge.

If the interaction model between domains has not been previously defined, the system (or each domain) must look for the needed information, analyzing where it is available or can be generated. The simplest knowledge source discovery mechanism is based on the traditional query/answer paradigm, where each part acts as client and server, interacting with other nodes directly sending queries or requests, and waiting until receiving an answer. This is only possible if the domains are previously known to each other or if a collaboration relationship has already been established. When this is not the case, the discovery of knowledge sources is affected by the dynamism of the system. Some nodes join and another ones leave the network, at any time. Besides, each domain is responsible for its own knowledge representation and management, because there are no a-priori agreements regarding ontology language nor granularity. This is the case of the Semantic Web [3].

Searching on the Semantic Web differs in several aspects from a traditional Web search, specially because of the structure of an online collection of documents in the Semantic Web, which consists of much more than HTML pages. The semantics associated to the languages for the Semantic Web allows the generation of new facts from existing facts, while traditional databases just enumerate all available facts. Traditional search machines do not try to understand the semantics of the indexed documents, while agents in the Semantic Web must do it [13]. In this work we define an ontology agent server for the Semantic Web, according to FIPA standards [8], and we named it Knowledge Source Discovery (KSD) agent. This agent is aware of which domains can provide knowledge in a specific area of knowledge and it can indicate a route to mobile agents carrying a user request. The agent we propose just knows the location (i.e. the url) of domains that can provide knowledge, but it does not provide the knowledge, nor analysis what the domain contains (i.e. files, pictures, documents, etc.). The KSD agent identifies the sources of information, but it is a task for the mobile agents to actually retrieve the knowledge (or the resources that contain it) from a domain. The architecture of a system that shows how this agent would interact with mobile agents travelling the Semantic Web is shown in figure 1. In the figure, a user generates a request (1), a mobile agent carries it (2) to the KSD agent, that provides an list of domains (3) that can reply it, the mobile agent follows the list (4) visiting the targeted domains, obtaining a response (5) that is finally presented to the user (6).

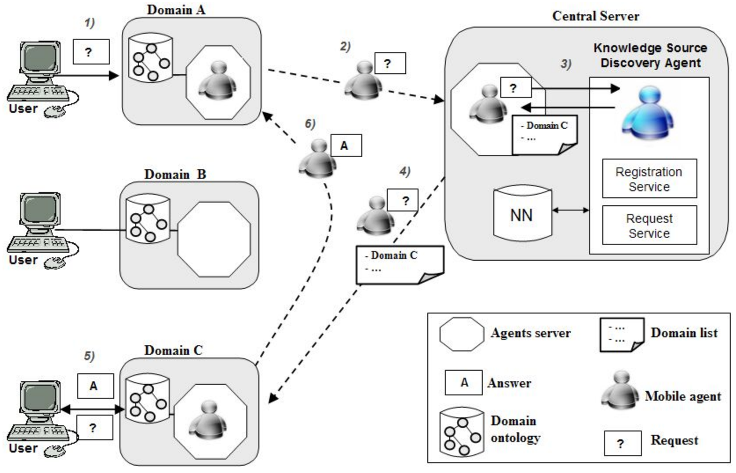

Fig. 1. A reference architecture for discovering distributed knowledge sources in the Semantic Web.

One of the KSD agent responsibilities is understanding (probably different) domain ontologies. In this scenario, each domain has its own ontology representing the knowledge it has, and an agent that identifies information sources should be able to identify dynamically which domains could satisfy a request brought to it by a mobile agent. This dynamic knowledge source discovery requires models and techniques that allow to find ontology concepts that have semantic affinity among them, even when they are different syntactically. As a first approach to this ontology-matching problem, in this paper we propose the use of computational intelligence tools. As a recent state-of-the-art review shows [9], Artificial Neural Network (ANN) models have been recently introduced for knowledge discovery in the Semantic Web. The review states that, generally speaking, ANNs provide a convenient way to represent knowledge for information retrieval applications, because, typically, the neurons may represent concepts and the connections among them, their semantic associations. In [15] it is shown how knowledge representation based on neural semantic networks is a natural way to simulate the human thinking ability, in words and entities, resulting in an intuitive and explicit representation of relations between concepts, in such a way that an agent can learn them.

2.1 Related work

In this paper, we have focused only on giving support for potential partners search on the Web, considering the existence of webpage metadata, which is similar to the approach proposed in the Helios Project [4], in which the problem of dynamic knowledge discovery in open distributed contexts has been addressed. Examples of open distributed contexts are Semantic Grids and Peer-based systems, where a set of independent peer nodes without prior reciprocal knowledge and no degree of relationship, dynamically need to cooperate by sharing their resources (such as data, documents or services). However, in contrast to our approach, the authors assume that no centralized authority manages a comprehensive view of the resources shared by all the nodes in the system, due to the dynamics of the collaborations and variability of the requirements.

The problem of answering queries considering metadata has been studied also in Peer-based systems, mainly addressing the problem of routing queries over a network. For example, Edutella [12] provides an infrastructure for sharing metadata in RDF format. The network is segmented into thematic clusters. In each cluster, a mediator semantically integrates source data. A mediator handles a request either directly or indirectly: directly, by answering queries using its own integrated schema; indirectly, by querying other cluster mediators by means of a dialog-based query processing module. Our approach shares the Edutella idea of implementing a mediator to guide the requests inside a specific context. However, in our proposal, the metadata is based on an ontology and each node could have its own ontology.

In the Semantic Web field, ANNs have been used to recognize automatically the connection between webpages with similar content and to improve semantic information retrieval [16] together with synonyms thesaurus, or based on text documents [14]. In a recent ontology-alignment state-of-the-art review [7] some tools based on ANNs are addressed, that using information regarding ontology schemas and instances, produce rules for ontology integration in heterogeneous databases. Our proposal is similarly to the work of [5], that uses instances to learn similarity between ontology concepts. However, differently from the previous approaches, we propose the use of a ANN model for helping the KSD decision process when a mobile agent carrying a request arrives, without the need of creating a newly integrated ontology to give a response. We propose an ANN model that can learn from several heterogeneous ontologies labels and instances, and is capable of matching a concept against the known heterogeneous ontologies.

3 The Knowledge Source Discovery Agent

To describe the operation of the KSD agent in an open distributed system, let us begin with a scenario where there is no such an agent. If this is the case, when a user of the system needs some information, he makes a request in natural language. We suppose that a mobile agent will be the responsible for finding a domain that could satisfy it. To do that, the mobile agent could visit all the domains it could find on the system, trying to find someone that could possibly match the request. An example: let us suppose a researcher from University A, wanting to collaborate with some other researcher from any other University, on the subject *Semantic Web*. The mobile agent carrying the request goes to any other Websites within the .edu domain, asking each one of them if they can reply to the request (or not). Besides time-wasting and performance problems, at this point two scenarios are possible: 1) all the domains use the same ontology, therefore the agent, querying the domain using RDQL can rapidly retrieve (positive or negative) an answer; 2) each domain has its own ontology and the mobile agent has to face an ontology-matching problem: it has simply nowhere to begin with, to establish if a domain can match the request or not. It could try asking a domain with some keywords from the request it carries, hoping that instances of the exactly same words appear somewhere on the domain Website. But most likely, this will not be the case.

For a scenario like 2), we introduce the KSD agent, whose responsibility is determining the sites that offer a greater possibility of providing the required information, and indicating the mobile agent which domains should be targeted. It handles the ontology-matching problem using machine learning techniques, which exploit the information contained in ontology instances. Whenever an information requirement is formulated from a domain, a mobile agent sends the request to the KSD agent. It reads and analyzes the request. This mechanism of processing the query in natural language will not be addressed in this paper. For all ontology concepts that must be matched, a classifier is built. The instances of each concept are presented to the ontology classifier, which decides whether there is a positive matching or not [6]. It then determines a list of domains that would provide the solicited information. To perform this task, the agent uses a ANN model with supervised learning, that allows it to guide queries in the specific R+D context.

4 The ontology-matching model for the KSD agent

We describe our understanding of *ontology-matching* [4] as: *Given a concept* c*, and two ontologies O_A and O_B, finding a match for the concept* c *means determining whether O_A and O_B (or both) have labels/instances with a semantic affinity to the target concept* c. Formally, we define our *ontology-matching* function as:

- *ontology-matching:*$c_i \rightarrow O_j$; i=1..n, j=A..Z.
- *ontology-matching(*c_1*)* $= \{O_A\}$, where c_1 is a concept related to O_A, which is a domain ontology related to the R+D field.

The central contribution of this paper is to present an approach for defining this matching function, using ANNs. Our proposal considers that labels at ontologies are human identifiers (names) for instances, normally shared by a community of humans speaking a common language inside a specific field of knowledge. We can, therefore, infer, that if labels are the same, the instances associated to them are probably also the same or are semantically related [6]. A concept, on the other hand, can also be defined as representative of a set of instances. We can, therefore infer, that concepts that have the same instances are the same. And viceversa, instances that have the same mother concept (label) are similar.

As previously stated, the agent is specialized in a determined field of knowledge (R+D domain). It has to be aware of all the domains related to this field. To do that, it periodically sends crawlers to index websites (like search engines do), or the domains can register with the agent because they want to be reachable. The KSD agent has a ANN model that is trained (and re-trained periodically) off-line with the data retrieved as stated in the previous paragraph. The model is trained with each domain ontology and their corresponding instances. Our basic assumption is that knowledge is captured in an arbitrary ontology encoding, based on a consistent semantics. From this, it is possible to derive additional knowledge such as, in our case, similarity of concepts in different ontologies. Understanding that labels describe concepts in natural language, one can derive that concepts/instances having the same labels are similar. This is not a rule which always holds true, but it is a strong indicator for similarity [6].

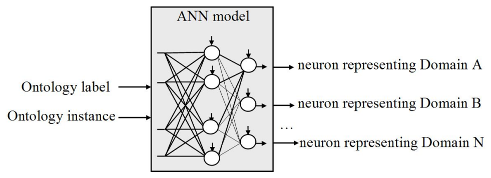
ANN model
Ontology label
Ontology instance
neuron representing Domain A
neuron representing Domain B
...
neuron representing Domain N

Fig. 2. ANN model for the KSD agent.

Our ANN is a multilayer perceptron model, which uses the standard back-propagation algorithm for supervised learning. That means that, for each input data point presented to the ANN model, there must be a corresponding matching output [10]. These {input/output} pairs are named training patterns. In our model, each training pattern is formed by: {*ontology-label,ontology-instance/Domain*} and a schematics representation of the model can be seen in figure 2. To explain the patterns formation for training the ANN model, we present a simple example. Let us suppose that we have three domains (A, B and C). Their corresponding domain ontologies and instances are shown in figure 3. As can be seen in the figure, each domain belongs to the R+D field of knowledge, but each one uses a different ontology to semantically annotate their websites. Domain A uses KA-ontology (*http://protege.cim3.net/file/pub/ontologies/ka*) provided by the free open source ontology editor *Protege*. Domain B uses the SWRC-ontology (*http://ontoware.org/projects/swrc*) (Semantic Web for Research Communities). Finally, domain C uses an own-highly-specialized ontology, in Spanish language.

Using the available data from all the domains (label names and associated instances), the ANN model training patterns are formed (see a) in figure 4). The labels and instances strings are codified using the standard ASCII code and are then normalized into the activation function domain of the hidden neurons, before entering the model, because this improves significantly training time and model accuracy. The ANN model parameters are set according to typical values. The number of input neurons for the neural model is set according to the maximum length of a combinations of label plus instance name, in any of the domain ontologies considered. The hidden layer neurons number is set empirically, with

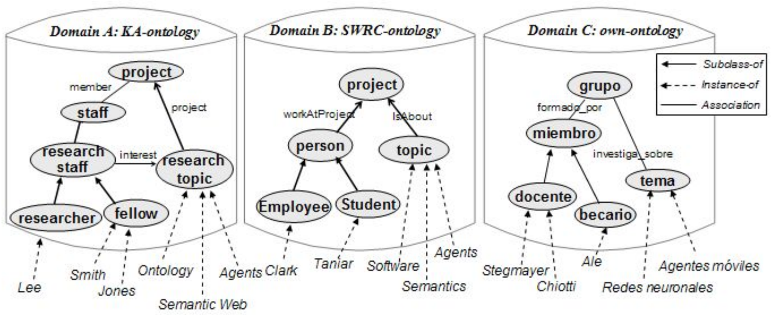

Fig. 3. Domain ontologies and instances.

linear activation functions. At the output, there is a specialized output neuron in the model for each domain, that recognizes when a domain ontology label or instance is presented to the model. The allowed values for each output neuron are 1 or 0, meaning that the neuron recognizes/does not recognizes a concept belonging to the domain the neuron represents.

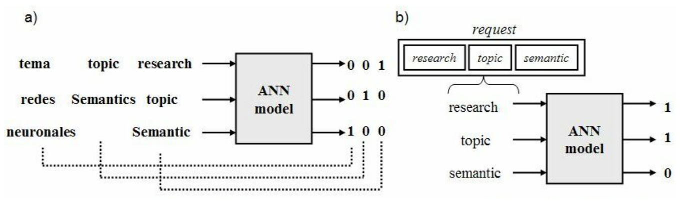

Fig. 4. a) Training the ANN model; b) Querying the ANN model.

4.1 ANN model use

The good generalization property of a network means the ability of a trained network to correctly predict a response to a set of inputs that has not seen before. It says that a trained network must perform well on a new dataset distinct from the one used for training. The KSD agent uses the neural model on-line, each time a mobile agent carrying a request "knocks on its door". The query is expressed in natural language by an end-user, and it is processed at the mobile agent, until only the keywords remain. These keywords are used to consult on the ANN model (see b) in figure 4). In the presented example, 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. Therefore, the KSD agent should return to the agent a Domain-list containing the domains A and B to be visited by the mobile agent. According to the ANN model of the KDS agent, those domains are very likely to be able to provide the required information. Because of how neural models work (interpolating data) and the properties associated to a three-layer multilayer perceptron model (generalization ability), the model would always provide a response [10]. Besides, standard learning algorithms can be used for the model, for which stability and convergence is guaranteed. If the request contains concepts totally unknown to the model, it shows that fact answering with high values, very different from zeros or ones.

4.2 Results

We have trained the ANN model with a small number of patterns (20) because, at the moment, we are working with small domains and with not highly populated ontologies. We have obtained an ANN model having 10 hidden neurons and a performance on the training patterns of 9E-02 after 10.000 iterations of the training algorithm. To measure the performance of the model, an information retrieval metric has been calculated over the training and test sets, obtaining 75% of *Recall*.

5 Discussion and future work

In this work we have presented an agent for Knowledge Source Discovery, responsible for guiding knowledge requirements towards distributed domains in the Semantic Web. It routes mobile agents through the web, indicating them the domains with the higher chances to reply to a request, using a neural network model. The ANN model is trained off-line with the knowledge contained in each domain, represented by ontology labels and instances. Each time a request arrives at the KSD agent, the ANN model is consulted regarding which domains could reply to the request. The reference architectural model for the agent, as well as some preliminary results on the ANN model recall performance, have been shown.

This is a first approach to the ontology-matching problem that an agent on the Semantic Web will have to face. We think that ANNs have the potential to solve this problem, probably with more powerful models, for example taking into account ontology relations, properties and constraints, instead of only labels and instances. We are also aware of the fact that, especially with big ontologies complexity of similarity calculations can grow exponentially. Already by comparing every entity with every other entity the approach becomes infeasible for big amounts of data. As data in ontologies expresses certain semantics the calculations might be channelled using these semantics e.g. starting with comparisons of top-level labels in the hierarchy. This is another direction of our future work. Furthermore, the KSD agent should also provide to the mobile agents arriving with requests, not only the lists of domains that could potentially response to the request, but also a way of interacting with the ontology domain. That is to say, if the query is expressed in English but the domain to visit is in Spanish, the mobile agent should have a mean (probably, the piece of ontology that could match his own) to talk to the domain ontology of the website providing answers.

References

1. R. Baeza-Yates, Web mining, in Proc. LA-WEB Congress (1): 2, 2005.
2. T. Berners-Lee, J. Hendler, O. Lassila. The Semantic Web, *Scientific American* 5(1): 29-37, 2001.
3. K. Breitman, M. Casanova, W. Truszkowski. *Semantic Web: Concepts, Technologies and Applications*, London: Springer-Verlag, 2007.
4. S. Castano, A. Ferrara, S. Montanelli. Dynamic Knowledge Discovery in Open, Distributed and Multi-Ontology Systems: Techniques and Applications, in: *Web Semantics and Ontology*, London: Idea Group Inc, 2006.
5. A. Chortaras et al. Learning Ontology Alignments Using Recursive Neural Networks, in Proc. Int. Conf. Neural Networks, Poland, (2)1: 811-816, 2005.
6. M. Ehrig, Y. Sure. Ontology Mapping - An Integrated Approach, in Proc. European Semantic Web Symposium, Greece,(1): 76-91, vol. 3053 LNCS-Springer, 2004.
7. J. Euzenat et al. State of the art on ontology alignment. D2.2.3, Technical Report IST-2004-507482, 2004. http://knowledgeweb.semanticweb.org
8. FIPA Ontology Service Specification, 2000. http://www.fipa.org
9. F. Gomide. *Enhancing the Power of Internet - Studies in Fuzziness and Soft Computing.* Heidelberg: Springer-Verlag, 2003.
10. S. Haykin. *Neural Networks*, New York: MacMillan, 1994.
11. J. Hendler. Agents and the Semantic Web, *IEEE Intelligent Systems* 16(2): 30-37, 2001.
12. W. Nejdl et al. EDUTELLA: A P2P Networking Infrastructure Based on RDF, in Proc. World Wide Web Conference, USA, 604-615, 2002.
13. E. Peis et al. Ontologías, metadatos y agentes: Recuperación semántica de la información, in Proc. Tratamiento y Recuperación de la Información, España, (1): 157-165, 2003.
14. S. Wermter. Neural Network Agents for Learning Semantic Text Classification, *Information Retrieval* 3(2): 87-103, 2000.
15. T. Zhang, S. Covaci. Adaptive Behaviors of Intelligent Agents based on Neural Semantic Knowledge, in Proc. Symp. Applications and the Internet, Japan, (1): 92-99, 2002.
16. X. Zhu et al. Recognizing the relations between Web pages using ANNs, in Proc. ACM Symposium on Applied Computing, USA, (1): 1217-1221, 2003.