Knowledge Modelling through RDF Classification

Vincenzo Loia, Sabrina Senatore and **Maria I. Sessa**
Dipartimento Matematica e Informatica - Universita' di Salerno
via Ponte Don Melillo - 84084 Fisciano (SA), Italy

Abstract

One of the most urgent problems presented on the web scenario is the difficulty to capture effective user-oriented information. Search engines return tons of data which often are so general as well as useless; sometimes the user request is so specific that no matches occur or they are too poor to satisfy the user expectation. Often the same information is recoverable from different web pages and documents, but in an imprecise or contradictory manner, so returned results are vague or disappointing. In last years, on the light of the emergency of the semantics, the tendency is to model meta-data about the web resources through RDF-based approaches which assure appreciable machine-oriented understandability. Objective of this work is framed in a wider project in-progress for semantic-based information discovery. Herein, in particular, we focus on the classification of RDF documents in order to capture semantics inside web resources as a valuable alternative to deal with the traditional content-based view of the web information.

Introduction

Although a lot of indexing and classification techniques exist, often the results returned by search engine are irrelevant, due to difficulty to capture the effective semantics related to user background which regularly does not meet the terms meaning indexed by search engines. In addition, search engines do not take into account the user preferences, replying the results according to own typical ranking (Kershberg, Kim, & Scime 2003). In reaction to this holdup, several research initiatives and commercial enterprises aim to improve existing information through machine-understandable semantics, in order to embrace all the web domains in the range from the knowledge management to electronic commerce. The Resource Description Framework (RDF) defines a syntactical data model for representing knowledge in a more structured form (resources as URI), in order to provide more expressiveness in modelling inference-based primitive through agent-oriented language (DARPA-DAML). On the other hand, because the HTML standard blends content and presentation into a single representation, end users are often influenced by the idea and vision of designer (Quan & Karger 2004). The Semantic Web, through RDF approach, eliminates these drawbacks, only accessing to pruned objective information. In this sense, the Semantic Web represents an extension of the Web where information assumes a well-defined meaning, in order to provide a universally accessible infrastructure which better works to guarantee more effective discovery, automation, integration and reuse across heterogeneous applications (Hendler, Berners-Lee, & Miller 2002). The RDF data model seems to overcome the restrictions of classical user-oriented HTML pages, where the human expressiveness, meaning vagueness and complexity of data source make more complicate right interpretation of user intents. The well-built RDF formalism and, at the same time, the flexible approach to the declarative representation of resources characterize the key direction to provide additional facilities in agent-based framework, in terms of rigorous parsing activities for capturing semantically meaningful data. Even though the RDF-based search tools are not so diffuse in the Web scenario, the present work proposes a semantic-based classification on a restricted (considering the limited impact on the Web) set of RDF pages. Our RDF classification points out the ontological aspect of data, rather than actual, human-oriented, content-based information. Purpose of this approach is to show how the RDF classification can give a proficient perspective of the conceptual information enclosed into RDF pages; in particular, through an appropriate user-driven selection of features, semantically related pages are clustered for reflecting the user's awareness and viewpoints. In the remainder of this paper, some preliminary considerations are exposed, in order to illustrate the whole architecture that embeds this framework; focusing on the core part, its main components are described, with the relative theoretical approach. Experiments and conclusions close the paper.

Preliminary overview

In (Loia *et al.* 2004) we presented an agent-based architecture for web searching; due to the absence of a standard in the construction of a HTML document, the interpretation of a web page was accomplished combining different granularity-based matching of web documents (such as

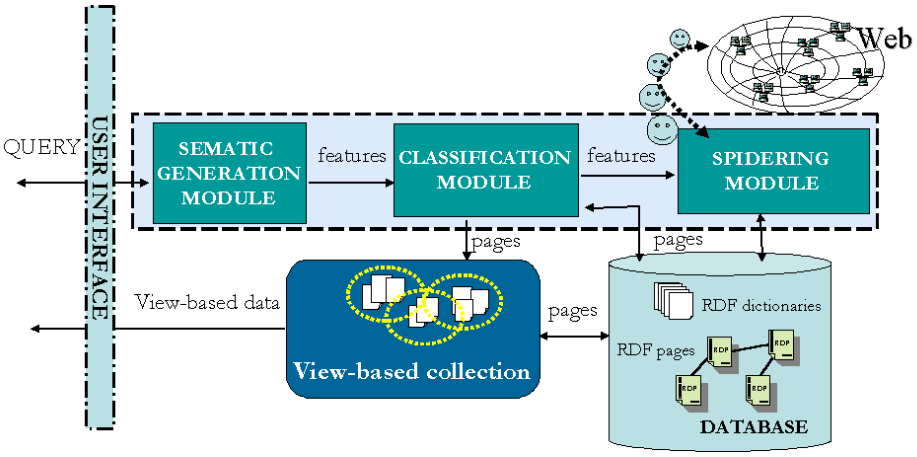

Figure 1: General View

structure, layout, content, etc). In response to the previous approach, the overall architecture shown in Figure 1 (actually in progress) represents a further step towards the design of a customized web searching, enriched with semantic-based resources. A combined framework is designed in order to couple the expressiveness power of semantic-oriented approach with a customized information retrieval. Objective is not to get back all pages related to requested information (that are often redundant) rather returning a related knowledge that assembles the more complete information inherent to input request. At the beginning, reflecting the user's intents, a refining process for depicting the right meaning of input query and its proper context is accomplished. Then, the gathered knowledge (concept) is employed to tailor the set of RDF pages which are closer to the request topic. So, the architecture of Figure 1 proposes an assisted RDF-based searching activity, finalized to customize the extracted information according to the user's acquaintance and requests. In particular, in this paper we focus on the core framework of presented architecture, designed to the classification of RDF documents.

Architecture

Figure 1 depicts the main modules (shown by square box) of the overall architecture, whose details are given in the following:

- *Semantic Generation Module*: produces an enriched concept, tailored on the user's input. Substantially, this module is based on the interaction with the linguistic database WordNet (Fellbaum C. 1998), that deals with sets of terms, *synsets*, in order to select the appropriate "word senses". By analyzing multiple sense of each input-given term, this module tries to elicit a richer sequence of words that better represents the given user's intent. This topic-oriented output (elicited concept) becomes the input of the *Classification Module* which will use it for the selection of feature terms.
- *Classification Module*: according to the computed feature space, a classification of RDF pages is returned. This module accomplishes two main activities: the *Clustering* activity carries out a clustering of the incoming RDF pages and the *Rule-based Classification* activity that, through fuzzy rules, classifies additional RDF pages (came from spidering activities), taking into account the previously performed clustering. Further details about this module are expanded in the following.
- *Spidering Module*: realizes a RDF spidering, in order to discover correlated argumentation, connected to query input. The spidering activity is realized by autonomous agents that are able to pull RDF resources from the analysis the RDF pages and the relative related RDF Schemas. Starting from references (links) discovered inside relevant classified RDF pages, the spider agents parse the reached pages and if the topic of search (terms of features) are encountered, send back the page to destination host, where it will be stored.

Working Flow

In order to provide a whole outlook, a sketch of global working flow is herein given. The input is a query compound by a word or a sequence of words describing a certain argument of interest. The *Semantic Generation Module* gets this input, filters (through WordNet) the multiple irrelevant meanings, focusing on the proper meaning and correlated sense (the multiplicity of term meanings is one of principal factors by which search engine returns inappropriate search results). The *Classification Module* takes as input these selected words (that better describe the given input) and the collection of the RDF pages, previously downloaded and stored in the database.

The use of user keywords (joined to some RDF metadata extracted by RDF pages) entails a clustering oriented to the user viewpoints. So, the *View-based Collection* of information is presented to user in a digest (interface-based) form. In particular, if the returned information is unsatisfactory (the user needs additional and complementary data), a spidering module is activated: starting from RDF pages, it launches spider agents on referenced links. The spiders

reach RDF-based pages that are parsed in order to decide if they are related to user topic. The parsing activity is accomplished through SiRPAC parser (W3C SiRPAC), distributed by W3C, in order to validate RDF/XML documents, on the basis of triple-based statements. Finally, collected pages are returned to the host and stored in the database for further data extraction analysis (through rule-based of *Classification Module*). At this stage, the system is able to return more exhaustive results, finalized to closely interpret the user intentions. Details about the *Classification Module* are given in the following; in fact the remainder of the paper formally details the two main activities (*Clustering* and *Rule-based Classification*) provided by this module.

Clustering of RDF pages

The clustering phase is accomplished by the well-known FCM algorithm (Bezdek, J.C. 1981), particularly useful for flexible data organization. Generally, FCM algorithm takes as input a data matrix where each row describes a RDF-*document* through a characteristic vector, built on a-priori fixed feature space. The features are represented by *terms*, obtained as combination of selected input words and RDF metadata, extracted by processed RDF collection of pages. In details, each row of *term-document* matrix is a weight-based vector which represents a RDF page $P \longleftrightarrow \underline{x} = (x_1, x_2, \ldots, x_n)$, where each component is a weight value associated to a computed term. Next section provides details about the selection of these relevant RDF metadata and the building of term-document matrix. After the FCM execution, partitions of RDF documents are returned, in a priori fixed number K of clusters.

Building of the term-document matrix

The building of term-document matrix is achieved through two focal steps. Firstly, in order to characterize the feature space, a selection procedure of relevant metadata in RDF pages is performed and then, for each RDF page, the vector-based representation is constructed.
The first step produces a selection of relevant metadata representing, at semantic level, the collection of RDF documents. Our analysis points on the RDF data which represent classes and properties rather than the relative assumed values. Recall that RDF approach defines each resource through triple-based statements: classes, properties and values. RDF pages depict information resources through RDF statements, on the basis of pre-constructed *definitions* of classes and properties (using the structure described by RDF Schema, ontologies, etc., declared inside the document). Bearing in mind that a RDF statement can be nested in other statement, in the calculus of relevant metadata, the level of nesting is considered. Possible candidate metadata are sought among class and property names.
In particular, the analysis of RDF documents highlights information based on different abstraction levels:

RDF dictionaries: all dictionaries (RDF Schemas and ontologies) declared inside a RDF document, in order to delineate the potential context of described information.

RDF metadata: as above described, let us consider *metadata* all terms (RDF tags) which are defined by dictionaries and used in the RDF document, surrounding the descriptive text. A parsing activity at this level, allows to individuate the conceptual context of examined RDF pages.

content of RDF tag: text included into RDF tags, representing the instances of resources defined into the dictionaries. This level gives an interpretation of effective data (values) associated to metadata.

In order to calculate the relevancy of metadata in RDF pages, let us give the following notation:

Collection of RDF pages: let P be the set of r RDF pages:

$$P = \{P_1, P_2, \ldots, P_r\}$$

Collection of schemas or dictionaries: Let D be the set of all schemas and - without loss of generality - dictionaries and ontologies used in the collection P:

$$D = \{D_1, D_2, \ldots, D_m\}$$

Dictionaries of the current RDF page: let P_i (with $1 \leq i \leq r$) be a generic page of collection P; the set of dictionaries of the page P_i, declared inside it, is:

$$DP_i = \{D_{i_1}, D_{i_2}, \ldots, D_{i_m}\}$$

where $D_{i_h} \in D$ for $1 \leq h \leq m$.

Each RDF page P_i depicts data through triple-based statements; each statement has a fixed structure, described by a resource (instance of a class which is defined in the dictionaries DP_i), a named property plus the value of that property for that resource (that can be a literal or another resource). Besides, more instances of the same class can exist inside the same RDF document and a statement can be composed of some nested statements.
So, for each RDF page P_i, represented by a sequence of statements, we evaluate only instances of class and the correlated properties.
Our approach computes, for each RDF instance, the degree of nesting of that instance in the statement and two measures (detailed in the sequel): the *accuracy* and *relevance*, associated to that instance. These measures, the relevance in particular, are estimated for all metadata encountered in each RDF document, in order to elicit appropriate candidates to represent features.

Accuracy: Fixed a class C of the dictionary DP_i, let us consider $A_1, A_2, \ldots, A_h$ (nested) statements in P_i, describing instances of C (that means the class name appears into the statement). Let $\pi(A_s, C)$ be a function which represents the number of distinct properties which describe the class C in the statement A_s. Thus we define the accuracy by the following formula:

$$Accuracy(A_s, C) = \frac{\pi(A_s, C)}{\sum_{j=1}^{h} \pi(A_j, C)} \qquad (1)$$

This value indicates the detail or granularity degree by which each instance of class is described in the RDF page.

Instance Relevance: it is a synthetic value representing the weight which the statement A_s assumed in the context of page P_i. It is computed as described in the following:

$$Inst\_Relevance(A_s, C) = \frac{Accuracy(A_s, C)}{nesting\ level} \quad (2)$$

This expression highlights the strong dependence between the nesting level where the instance statement is defined and the accuracy by which the instance of the considered class is described.

These measures compute the relevance associated to all statement in a RDF page; so for each page, information relative to all instances of class are collected. An analogous analysis is accomplished, considering properties associated to classes in the statements of a RDF page: each resource (instance) can have different properties or more occurrences of same property. So, for each (univocally identified) property, associated to an instance of a class (through a relative dictionary DP_i) the *relevance* value is computed. Now, for each RDF page P_i, let us define:

Property Relevance: let p be a property associated to an instance A_s of class C and let $Inst\_Relevance(A_s, C)$ be the previously computed parameter for a generic statement A_s; the relevance degree for the property p is so defined:

$$Prop\_Relevance(A_s, p) = Inst\_Relevance(A_s, C) \cdot \#p \quad (3)$$

where $\#p$ represents the number of occurrences of property p in the statement A_s. This expression highlights the dependence from the relevance of the instance to which the property is associated. In summary, two relevance measures are previously built: one binds instances and relative classes, another one is defined between the instances and relative properties. So, in each RDF page, some values of relevance are computed for each instances of same class and for each property related to same instance.
In order to elicit the appropriate features for building the term-document matrix, a digest (summarized) relevance value is computed for each metadata (class and property names) of that RDF page. So, given a page, for each class name C (of a RDF dictionary), the sum of all relevance values of all related instances is calculated. In the same way, the sum of relevance values on each property name p is computed. We call this relevance value *Metadata_Relevance(M)* where M represents a class C or a property p names. It represents the summarized weight (the relevance) of each distinct metadata, for each RDF page. Concluding, for each RDF page, a list of relevance-based values for each distinct metadata (discovered inside of it) is associated. Finally, in order to evaluate a global (summarized) relevance for each RDF metadata M, on the whole collection of RDF pages, the averages on all distinct *Metadata_Relevance(M)*, with respect to all given pages, are computed. In this way, a ranked list of relevance values is obtained, for each metadata defined into a dictionary DP of the whole collection P of RDF pages. Figure 2 gives a snapshot of an interactive interface which shows a ranked list of computed RDF metadata, candidate for the feature space's definition. At this point, the selected words of the input topic (returned by the *Semantic Generation Module*) are compared with the metadata in the list. Through WordNet, synonyms, hyponyms and other related words are considered to detect the common semantic-linked words, in order to elicit the actual, user-oriented list of most relevant metadata. The candidate features are the first α metadata (with $\alpha \in N$) in this ranked list. Once the features are selected, the term-document matrix can be defined: each row describes a RDF page through a vector, where each element is a value associated to a selected term (word or metadata) designed as features. In particular, each value of the vector just is the previously computed *Metadata_Relevance(M)* for the metadata M, normalized to 1.

Rule-based classification

This phase achieves a further classification of additional RDF documents, on the base of existing clusters. Through fuzzy rules generated by the RDF clustering, incoming RDF pages (discovered during the spidering activity) can be assigned to an existing class/cluster of a previous activity of clustering. This phase can be framed into a knowledge management approach (for example, a bookmarking-like activity) where RDF-oriented semantics enrich the local information base and refine the local RDF recognizance.

Rules generation

The collected pages are classified by means of the FCM algorithm; considering that the complexity of algorithm is proportional to data size, it is not practical to handle periodic updating of RDF collection (re-performing the FCM-based classification), whenever the spidering action returns discovered RDF pages. So, the new ingoing web pages are classified through fuzzy-based rules (Hoppner *et al.* 1999), defined on clusters of the FCM algorithm and obtained on the current RDF collection. In literature, fuzzy if-then rules are often used for this kind of task, because they well adapt themselves to provide a human description of structures in data (Klose A. 2003).

Recall each page P is given in input to FCM as a vector $\underline{x} = (x_1, x_2, \ldots, x_n) \in [0,1]^n$, where each $x_j, j = 1, \ldots, n$ represents the weight value (or normalized relevance) of the jth feature (or metadata) evaluated on the page P. The FCM execution produces a partition of the data matrix in (a-priori fixed) number K of clusters. According to the cylindrical extension (a projection-based method of n-dimensional argument vector: see Hoppner *et al.* 1999), a generic fuzzy cluster K_i with $i = 1, \ldots, K$ can be described through the following membership functions:

$$\mu_{i_1}(x_1),\ \ \mu_{i_2}(x_2),\ \ \ldots\ \ ,\mu_{i_n}(x_n)$$

that represent the membership degrees of each component of vector $\underline{x}$ in the ith cluster K_i. The objective is to obtain a unique membership value of vector for this cluster;

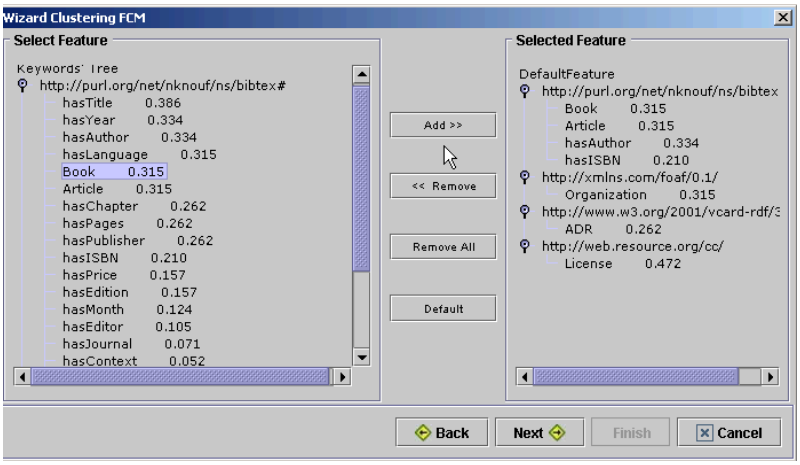

Figure 2: Ranked list of computed terms for the features selection

exploiting the conjunction (evaluated in *and*) of the projection of n-dimensional argument vector (used as the scalar argument of membership function in the projection space) (Hoppner *et al.* 1999) the evaluation of membership degree $\mu_i(\underline{x})$ of a vector $\underline{x}$ in the ith cluster K_i with $i = 1, \ldots, K$, can be defined:

$$\mu_{i_1}(x_1) \ and \ \mu_{i_2}(x_2) \ and \ \ldots \ and \ \mu_{i_n}(x_n)$$

It simply follows that the fuzzy rule associated for the ith class (thanks to an assignment of linguistic label to projected membership function, clusters can be interpreted as classes) is the following:

If $(x_1 \ is \ \mu_{i_1})$ *and* $(x_2 \ is \ \mu_{i_2})$ *and*$\ldots$ *and* $(x_n \ is \ \mu_{i_n})$
then $\underline{x}$ *is in the class/cluster* K_i *with* $\mu_i(\underline{x})$

where the conjunction is evaluated by the minimum.
The rule expresses the relation which exists between the membership functions assumed by characteristic vector (evaluate in the antecedent rule) and the global membership value for the ith class.
Then the generic page, depicted as vector $\underline{x}$ is assigned to class K_j such that:

$$\mu_j(\underline{x}) = \max_{1 \leq i \leq n} \mu_i(\underline{x})$$

Figure 3 provides a screenshot which graphically shows the membership function associated to each feature in the clusters. In details, fuzzy rules enable to assign incoming RDF pages to defined (FCM-based) classes; sometimes, some misclassifications can occur because new pages could not accurately be assigned to classes. There are two possibilities to consider: the information are not sufficient to classified them (low membership values for each class) or the classification can introduce an approximation error. These pages become "undecided" due to fact that more information (based on different or further features) occurs to correctly classified them so new classification phase is necessary.

Experimental Results

Although the heterogeneity of web resources proclaims the exigency of machine-processable information, the diffusion of semantic, machine-readable applications is yet limited. This approach can be framed in Virtual Organizations or intranet-based activities, where, according to profiles of employees, different user-driven characterizations of information can be tailored. In our approach, a user interrogates the system, through a query based on specific resources-oriented data, according to own viewpoints. A valuation of system performance was realized on a set of 50 RDF pages, 35 of them downloaded by FOAF group (FOAF), the remaining are created using known RDF schemas (foaf, vs, wot, etc.). The pages are selected to represent a restricted number of topic categories, according to the number of clusters, fixed a priori. We sketch three meaningful experiments in order to show the performance of this approach. Table 1 shows the results. Each experiment is composed of two phases: one evaluates only the returned information after the classification, the other phase, instead, through the spidering activity, achieves the classification of incoming pages, by fuzzy rules. Experiment 1 considers 7 features, 3 of them are related to the input topic. Both results present a similar (actual) recall value and a quite high error of evaluation that represents the percentage of unwanted (not relevant) results. The results based on fuzzy rules (Experiment 1 with FR), classify well all five incoming pages, producing the same error percentage. Similar considerations are feasible for the experiment 2 where the features set is bigger, with 4 terms belonging to user's request. In this case, the error is smaller in the rule-based approach, although the expected recall is higher (some new pages are classified, others are not). Finally experiment 3 with 13 features (4 of them related to query input) shows a discrete performance in the rule-based approach (returned pages do not influence the number of unwanted pages). As possible extension, further improvements will be considered to enrich the feature space of term-document matrix which represents the focal point for increasing the system performance.

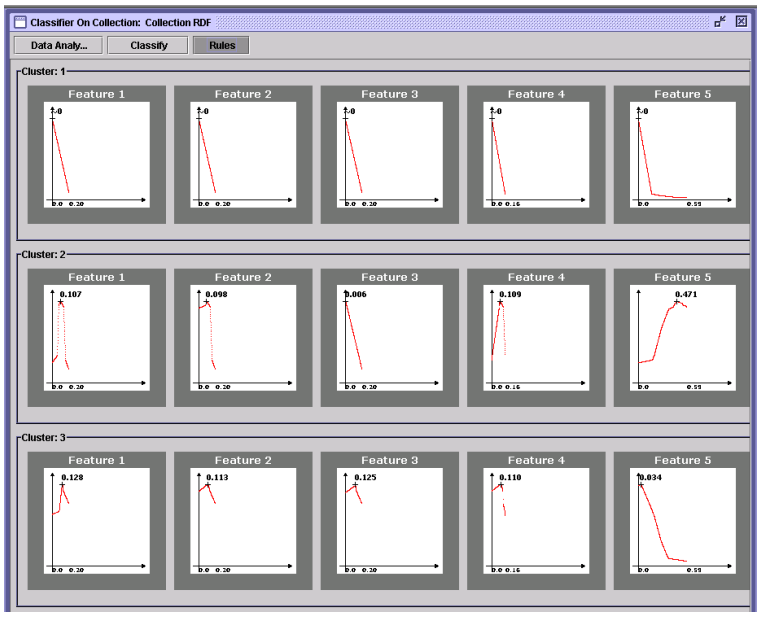

Figure 3: Rules extraction by features analysis

Experim.	features	waiting recall	actual recall	error
1	7	70%	60%	11%
1 with FR	7	73%	62%	11%
2	10	87%	71%	6%
2 with FR	10	88%	71%	5%
3	13	87%	77%	4%
3 with FR	13	90%	80%	4%

Table 1: Performances expressed in terms of recall and relative error (percentage).

Conclusions

The popularity of the Web as a diffusion medium of knowledge and technology engulf the network continuously, with new resources such as new data, new services, hypermedia. Looking at current web scenario, a profusion of heterogeneous information confuses users: discovering valuable information (for given criteria) becomes a tiring and often unproductive task, on the light of the tons of junk and redundant information. Although most of web documents are human-oriented (based on HTML standard), today the semantic-oriented approach is assuming a prominent role in the understandability of the ontological aspects embedded into Web pages (Rocha, Schwabe, & Poggi de Aragão 2004; Knublauch 2003), considering the development of machine-oriented technologies (agent-based systems) for surfing the Web space and carrying out tasks for humans. The present approach moves ahead in this direction, focusing on a semantic-based classification of RDF documents as possible contribution towards a more formal characterization of relationships between a concept and a web resource.

References

Bezdek, J.C. 1981. *Pattern Recognition and Fuzzy Objective Function Algorithms*. N. York: Plenum Press.

DARPA-DAML. DARPA Agent Markup Language. http://www.daml.org/.

Fellbaum C. 1998. WordNet An Electronic Lexical Database.

FOAF. The Friend of a Friend project. http://www.foaf-project.org/.

Hendler, J.; Berners-Lee, T.; and Miller, E. 2002. Integrating applications on the semantic web. *Institute of Electrical Engineers of Japan* 122(10):676–680.

Hoppner, F.; Klawonn, F.; Kruse, R.; and Runkler, T. 1999. *Fuzzy Cluster Analysis - Methods for Image Recognition*. N. York: J. Wiley.

Kershberg, L.; Kim, W.; and Scime, A. 2003. A presonalized agent for Semantic Taxonomy-Based Web Search. *Electronic Commerce Research and Applications* 1(2).

Klose A. 2003. Extracting fuzzy classification rules from partially labeled data. *Soft Computing Springer-Verlag*.

Knublauch, H. 2003. An AI tool for the real world - knowledge modeling with protege. *Java World*. Walkthrough of Protege.

Loia, V.; Pedrycz, W.; Senatore, S.; and Sessa, M. 2004. Proactive utilization of proximity-oriented information inside an Agent-based Framework. In *17th International FLAIRS Conference FLAIRS 2004*.

Quan, D., and Karger, D. 2004. How to Make a Semantic Web browser. In *Proceedings of WWW 2004*. ACM Press.

RDF. Resource Description Framework. http://www.w3.org/RDF/.

Rocha, C.; Schwabe, D.; and Poggi de Aragão, M. 2004. A hybrid approach for Searching in the Semantic Web. In *Proceedings of WWW 2004*, 374–383. ACM Press.

W3C SiRPAC. A simple RDF parser and compiler.