Improving Ontology Recommendation and Reuse in WebCORE by Collaborative Assessments

Iván Cantador, Miriam Fernández, Pablo Castells

Escuela Politécnica Superior
Universidad Autónoma de Madrid
Campus de Cantoblanco, 28049, Madrid, Spain

{ivan.cantador, miriam.fernandez, pablo.castells}@uam.es

ABSTRACT

In this work, we present an extension of CORE [8], a tool for Collaborative Ontology Reuse and Evaluation. The system receives an informal description of a specific semantic domain and determines which ontologies from a repository are the most appropriate to describe the given domain. For this task, the environment is divided into three modules. The first component receives the problem description as a set of terms, and allows the user to refine and enlarge it using WordNet. The second module applies multiple automatic criteria to evaluate the ontologies of the repository, and determines which ones fit best the problem description. A ranked list of ontologies is returned for each criterion, and the lists are combined by means of rank fusion techniques. Finally, the third component uses manual user evaluations in order to incorporate a human, collaborative assessment of the ontologies. The new version of the system incorporates several novelties, such as its implementation as a web application; the incorporation of a NLP module to manage the problem definitions; modifications on the automatic ontology retrieval strategies; and a collaborative framework to find potential relevant terms according to previous user queries. Finally, we present some early experiments on ontology retrieval and evaluation, showing the benefits of our system.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *information filtering, retrieval models, selection process*.

General Terms

Algorithms, Measurement, Human Factors.

Keywords

Ontology evaluation, ontology reuse, rank fusion, collaborative filtering, WordNet.

1. INTRODUCTION

The Web can be considered as a live entity that grows and evolves fast over time. The amount of content stored and shared on the web is increasing quickly and continuously. The global body of multimedia resources on the Internet is undergoing a significant growth, reaching a presence comparable to that of traditional text contents. The consequences of this enlargement result in well known difficulties and problems, such as finding and properly managing all the existing amount of sparse information.

To overcome these limitations the so-called "Semantic Web" trend has emerged with the aim of helping machines process information, enabling browsers or other software agents to automatically find, share and combine information in consistent ways. As put by Tim Berners-Lee in 1999, "*I have a dream for the Web in which computers become capable of analyzing all the data on the Web – the content, links, and transactions between people and computers. A 'Semantic Web', which should make this possible, has yet to emerge, but when it does, the day-to-day mechanisms of trade, bureaucracy and our daily lives will be handled by machines talking to machines. The 'intelligent agents' people have touted for ages will finally materialize*".

At the core of these new technologies, ontologies are envisioned as key elements to represent knowledge that can be understood, used and shared among distributed applications and machines. However, ontological knowledge mining and development are difficult and costly tasks that require major engineering efforts. Developing an ontology from scratch requires the expertise of at least two different individuals: an ontology engineer that ensures the correctness during the ontology design and development, and a domain expert, responsible for capturing the semantics of a specific field into the ontology. In this context, ontology reuse becomes an essential need in order to exploit past and current efforts and achievements.

In this scenario, it is also important to emphasize that ontologies, as well as content, do not stop evolving and growing within the Web. They are part of its wave of growth and evolution, and they need to be managed and kept up to date in distributed environments. In this perspective, the initial efforts to collect ontologies in libraries [17] are not sufficient, and novel technologies are necessary to successfully retrieve this special kind of content.

Novel tools have been recently developed, such as ontology search engines [24] represent an important first step towards automatically assessing and retrieving ontologies which satisfy user queries and requests. However, ontology reuse demands additional efforts to address special needs and requirements from ontology engineers and practitioners. It is necessary to evaluate and measure specific ontology features, such as lexical vocabulary, relations [11], restrictions, consistency, correctness, etc., before making an adequate selection. Some of these features can be measured automatically, but some, like the correctness or the level of formality, require a human judgment to be assessed.

In this context, the Web 2.0 is arising as a new trend where people collaborate and share their knowledge to successfully achieve their goals. New search engines like Technorati[1] exploit blogs with the aim of finding not only the information that the user is looking for, but also the experts that might better answer the users' requirements. As put by David Sifry, one of the founders of

[1] Technorati, blog search engine, http://technorati.com/

Technorati, in an interview for a Spanish newspaper, "*Internet has been transformed from the great library to the great conversation*".

Following this aspiration, the work presented here aims to enhance ontology retrieval and recommendation, combining automatic evaluation techniques with explicit users' opinions and experiences. This work follows a previous approach for Collaborative Ontology Reuse and Evaluation over controlled repositories, named CORE [8]. For the work reported in this paper, the tool has been enhanced and adapted to the Web. Novel technologies, such as AJAX[2], have been incorporated to the system for the design and implementation of the user interface. It has also been modified and improved to overcome previous limitations, such as handling large numbers of ontologies. The collaborative capabilities have also been extended within two different frameworks. Firstly, during the problem definition phase, the system helps users to express their needs and requirements by showing other problem descriptions previously given by different users. Secondly, during the ontology retrieval phase, the system helps users to enhance the automatic system recommendations by using other user evaluations and comments.

Following Leonardo Da Vinci's words, "*Wisdom is the daughter of experience*", our tool aims to take a step forwards for helping users to be wise in exploiting other people's experience and expertise.

The rest of the paper has been organized as follows. Section 2 summarizes some relevant work related to our system. Its architecture is described in Section 3. Section 4 contains empirical results obtained from early experiments done with a prototype of the system. Finally, several conclusions and future research lines are given in Section 5.

2. RELATED WORK

2.1 Ontology Evaluation

Two well-known scenarios for ontology reuse have been identified in the Semantic Web area. The first one addresses the common problem of finding the most adequate ontologies for a specific domain. The second scenario envisions the not so common but real situation in which Semantic Web applications need to automatically and dynamically find an ontology. In this work, we focus our attention on the fist scenario, where users are the ones who express their information needs. In this scenario, ontology reuse involves several areas such as ontology evaluation, selection, search and ranking.

Several ontology libraries and search engines have been developed in the last few years to address the problem of ontology search and retrieval. [6] presents a complete study of ontology libraries (WebOnto, Ontolingua, SHOE, etc.), where their functionalities are evaluated attending to different criteria such as ontology management, ontology adaptation and ontology standardization. Although ontology libraries are a good temporary solution for ontology retrieval, they suffer from the current limitation of not being opened to the web. In that sense, Swoogle [24] constitutes one of the biggest efforts carried out to crawl, index and search for ontologies distributed across the Web.

To obtain the most appropriate ontology and fulfil ontology engineers' requirements, search engines and libraries should be complemented with evaluation methodologies.

Ontology evaluation can be defined as assessing the quality and the adequacy of an ontology for being used in a specific context, for a specific goal. From our perspective, ontology evaluation constitutes the cornerstone of ontology reuse because it faces the complex task of evaluate, and consequently select the most appropriate ontology on each situation.

An overview of ontology evaluation approaches is presented in [4], where four different categories are identified: those that evaluate an ontology by comparing it to a Golden Standard [11]; those that evaluate the ontologies by plugging them in an application and measuring the quality of the results that the application returns [16]; those that evaluate ontologies by comparing them to unstructured or informal data (e.g. text documents) [5], and those based on human interaction to measure ontology features not recognizable by machines [10]. In each of the above approaches several evaluation levels are identified: lexical, taxonomical, syntactic, semantic, contextual, and structural between others. Table 1 summarized these ideas.

Table 1. An overview of approaches to ontology evaluation

	Approach			
Level	*Golden Standard*	*Application based*	*Data driven*	*Assessment by humans*
Lexical entries, vocabulary, concept, data	X	X	X	X
Hierarchy, taxonomy	X	X	X	X
Other semantic relations	X	X	X	X
Context, application		X		X
Syntactic	X			X
Structure, architecture, design				X

Once the ontologies have been searched, retrieved and evaluated, the next step is to select the most appropriate one that fulfils user or application goals. Some approaches for ontology selection have been addressed in [20] and complemented in [19], where a complete study is presented to determine the connections between ontology selection and evaluation.

When the user and not the application is the one that demands an ontology, the selection task should be less categorical, returning not only one but the set of the most suitable results. To sort these results according to the evaluation criteria, several ontology ranking measures have been proposed in the literature. Some of them are presented in [2] and [3]. Both works aim to take a step beyond to the approaches based on the page-rank algorithm [24], where ontologies are ranked considering the number of links between them, because this ranking methodology does not work for ontologies with poor connectivity and lack of referrals from other ontologies.

[2] Garrett, J. J. (2005). *AJAX: A New Approach to Web Applications.* In http://www.adaptivepath.com/

As it has been shown before, current ontology reuse approaches take advantage of ontology evaluation, search, retrieval, selection and ranking methodologies. All these areas provide different advantages to the process of ontology evaluation and reuse, but they do not exploit others related to the well known Recommender Systems [1]; is it helpful to know other users' opinions to evaluate and select the most suitable ontology?

The collaboration between users has been addressed in the area of ontology design and construction [23]. In [14], the necessity of mechanisms for ontology maintenance is presented under scenarios like "ontology-development in collaborative environments". Moreover, works as [7], present tools and services to support the process of achieving consensus on common shared ontologies by geographically distributed groups.

However, despite all these common scenarios where the user's collaboration is required for ontology design and construction, the use of collaborative tools for ontology evaluation is still a novel and incipient approach in the literature [8].

2.2 Recommender Systems

Collaborative filtering strategies make automatic predictions (filter) about the interests of a user by collecting taste information from many users (collaborating). This approach usually consists of two steps: a) look for users that have a similar rating pattern to that of the active user (the user for whom the prediction is done), and b) use the ratings of users found in the previous step to compute the predictions for the active user. These predictions are specific to the user, differently to those given by more simple approaches that provide average scores for each item of interest, for example based on its number of votes.

Collaborative filtering is a widely explored field. Three main aspects typically distinguish the different techniques reported in the literature [13]: user profile representation and management, filtering method, and matching method.

User profile representation and management can be divided into five different tasks:

- *Profile representation*. Accurate profiles are vital for the content-based component (to ensure recommendations are appropriate) and the collaborative component (to ensure that users with similar profiles are in fact similar). The type of profile chosen in this work is the user-item ratings matrix (ontology evaluations based on specific criteria).

- *Initial profile generation*. The user is not usually willing to spend too much time in defining her/his interests to create a personal profile. Moreover, user interests may change dynamically over time. The type of initial profile generation chosen in this work is a manual selection of values for only five specific evaluation criteria.

- *Profile learning*. User profiles can be learned or updated using different sources of information that are potentially representative of user interests. In our work, profile learning techniques are not used.

- *The source of user input and feedback* to infer user interests from information used to update user profiles. It can be obtained in two different ways: using information explicitly provided by the user, and using information implicit observed in the user's interaction. Our system uses no feedback to update the user profiles.

- *Profile adaptation*. Techniques are needed to adapt the user profiles to new interests and forget old ones as user interests evolve with time. Again, in our approach profile adaptation is done manually (manual update of ontology evaluations).

Filtering method. Items or actions are recommended to a user taking into account the available information (item content descriptions and user profiles). There are three main information filtering approaches for making recommendations:

- *Demographic filtering*: Descriptions of people (e.g. age, gender, etc) are used to learn the relationship between a single item and the type of people who like it.

- *Content-based filtering*: The user is recommended items based on the descriptions of items previously evaluated by other users. Content-based filtering is chosen approach in our work (the system recommends ontologies using previous evaluations of those ontologies).

- *Collaborative filtering*: People with similar interests are matched and then recommendations are made.

Matching method. It defines how user interests and item characteristics are compared. Two main approaches can be identified:

- *User profile matching*: people with similar interests are matched before making recommendations.

- *User profile-item matching*: a direct comparison is made between the user profile and the items. The degree of appropriateness of the ontologies is computed by taking into account previous evaluations of those ontologies.

In WebCORE, a new ontology evaluation measure based on collaborative filtering is proposed, considering users' interests and previous assessments of the ontologies.

3. SYSTEM ARCHITECTURE

As mentioned before, WebCORE is a web application for Collaborative Ontology Reuse and Evaluation. A user logins into the system via a web browser, and, thanks to AJAX technology and the Google Web Toolkit[3], dynamically describes a problem domain, searches for ontologies related to this domain, obtains relevant ontologies ranked by several lexical, taxonomic and collaborative criteria, and optionally evaluates by himself those ontologies that he likes or dislikes most.

In this section, we describe the server-side architecture of WebCORE. Figure 1 shows an overview of the system. We distinguish three different modules. The first one, the left module, receives the problem description (Golden Standard) as a full text or as a set of initial terms. In the first case, the system uses a NLP module to obtain the most relevant terms of the given text. The initial set of terms can also be modified and extended by the user using WordNet [12]. The second one, represented in the centre of the figure, allows the user to select a set of ontology evaluation techniques provided by the system to recover the ontologies closest to the given Golden Standard. Finally, the third one, on the right of the figure, is a collaborative module that re-ranks the list of recovered ontologies, taking into consideration previous feedback and evaluations of the users.

[3] Google Web Toolkit, http://code.google.com/webtoolkit/

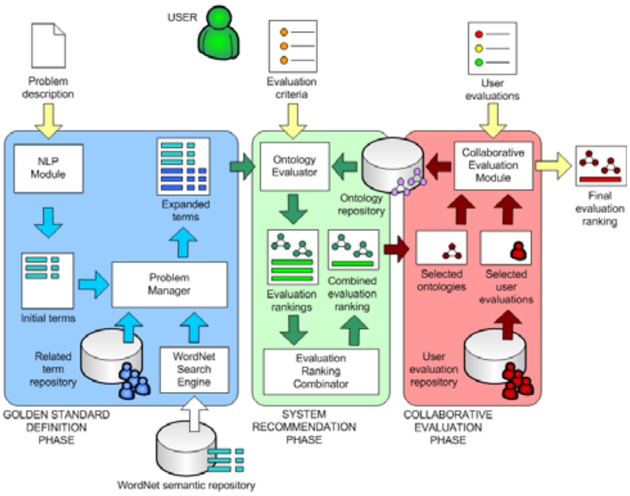

USER
Problem description
Evaluation criteria
User evaluations
NLP Module
Expanded terms
Initial terms
Problem Manager
Related term repository
WordNet Search Engine
Ontology Evaluator
Ontology repository
Evaluation rankings
Combined evaluation ranking
Evaluation Ranking Combinator
Collaborative Evaluation Module
Selected ontologies
Selected user evaluations
User evaluation repository
Final evaluation ranking
GOLDEN STANDARD DEFINITION PHASE
SYSTEM RECOMMENDATION PHASE
COLLABORATIVE EVALUATION PHASE
WordNet semantic repository

Figure 1. WebCORE architecture

3.1 Golden Standard Definition

The first phase of our ontology recommender system is the Golden Standard definition. As done in the first version of CORE [8], the user describes a domain of interest specifying a set of relevant terms that will be searched through the concepts (classes or instances) of the ontologies stored in the system.

As an improvement, WebCORE includes an internal NLP component that automatically retrieves the most informative terms from a given text. Moreover, we have added a new collaborative component that continuously offers to the user a ranked list with the terms that have been used in those previous problem descriptions in which a given term appears.

3.1.1 Term-based Problem Description

In our system, the Golden Standard is described by a set of initial set of terms. These terms can automatically be obtained by the internal Natural Language Processing (NLP) module, which uses a repository of documents related to the specific domain in which the user is interested in. This NLP module accesses to the repository of documents, and returns a list of pairs (lexical entry, part of speech) that roughly represents the domain of the problem. On the other hand, the list of initial (root) terms can be manually specified.

The module also allows the user to expand the root terms using WordNet [12] and some of the relations it provides: hypernym, hyponym and synonym. The new terms added to the Golden Standard using these relations might also be extended again, and new terms can iteratively be added to the problem definition.

The final representation of the Golden Standard is defined as a set of terms T (L^G, POS, L^{GP}, R, Z) where:

- L^G is the set of lexical entries defined for the Golden Standard.
- POS corresponds to the different Parts Of Speech considered by WordNet: *noun*, *adjective*, *verb* and *adverb*.
- L^{GP} is the set of lexical entries of the Golden Standard that have been extended.
- R is the set of relations between terms of the Golden Standard: *synonym*, *hypernym*, *hyponym* and *root* (if a term has not been obtained by expansion, but is one of the initial terms).
- Z is an integer number that represents the depth or distance of a term to the root term from which it has been derived.

Examples:

T_1 = ("genetics", NOUN, "", ROOT, 0). T_1 is one of the root terms of the Golden Standard. The lexical entry that it represents is "genetics", its part of speech is "noun", it has not been expanded from any other term so its lexical parent is the empty string, its relation is "root", and its depth is 0.

T_2 = ("biology", NOUN, "genetics", HYPERNYM, 1). T_2 is a term expanded from "genetics" (T_1). The lexical entry it represents is "biology", its part of speech is "noun", the lexical entry of its parent is "genetics", it has been expanded by the "hypernym" relation, and the number of relations that separates it from the root term T_1 is 1.

Figure 2 shows the interface of the Golden Standard Definition phase. In the left side of the screen, the current list of root terms is shown. The user can manually insert new root terms to this list giving their lexical entries and selecting their parts of speech. The correctness of these new insertions is controlled by verifying that all the considered lexical entries belong to the WordNet repository. Adding new terms, the final Golden Standard definition is immediately updated: the final list of (root and expanded) terms that represent the domain of the problem is shown in the bottom of the figure. The user can also make term expansion using WordNet. He selects one of the terms from the Golden Standard definition and the system shows him all its meanings contained in WordNet (top of the figure). After he has chosen one of them, the system presents him three different lists with the synonyms, hyponyms and hypernyms of the term. The user can then selects one or more elements of these lists and add them to the expanded term list. For each expansion, the depth of the new term is increased by one unit. This will be used later to measure the importance of the term within the Golden Standard: the greater the depth of the derived term with respect to its root term, the less its relevance will be.

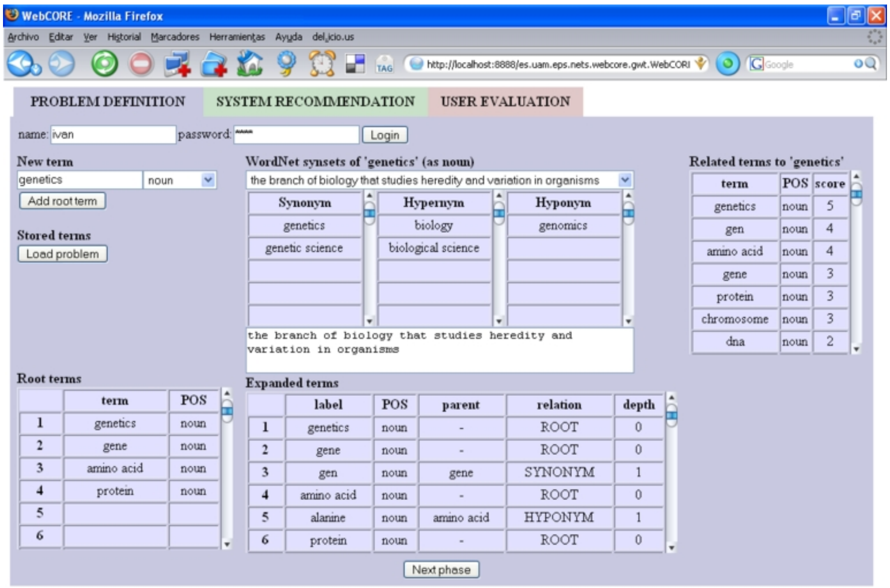
3.1.2 Collaborative Problem Description

In the problem definition phase a collaborative component has been added to the system (right side of Figure 2). This component reads the term currently selected by the user, and searches for all the stored problem definitions that contain it. For each of these problem definitions, the rest of their terms and the number of problems in which they appear are retrieved and shown in the web browser.

With this simple strategy the user is suggested the most popular terms, fact that could help him to better describe the domain in which he is interested in. It is very often the case that a person has very specific goals or interests, but does not know how to correctly explain/describe them, and how to effectively find solutions for them. With the retrieved terms, the user might discover new ways to describe the problem domain and obtain better solutions in the ontology recommendation phase.

This follows somehow the ideas of the well known folksonomies[4]. The term "folksonomy" is a combination of "folk" and "taxonomy", and was firstly used by Thomas Vander Wal [22] in

[4] Mathes, A. (2004). *Folksonomies: Cooperative Classification and Communication through Shared Metadata*. http://www.adammathes.com/academic/computer-mediated-communication/folksonomies.html


WebCORE - Mozilla Firefox
Archivo Editar Ver Historial Marcadores Herramientas Ayuda del.icio.us
http://localhost:8888/es.uam.eps.nets.webcore.gwt.WebCORI
Google
PROBLEM DEFINITION
SYSTEM RECOMMENDATION
USER EVALUATION
Login
Add root term
Load problem
Next phase

name: ivan password: ****

New term

genetics noun

Stored terms

WordNet synsets of 'genetics' (as noun)

the branch of biology that studies heredity and variation in organisms

Synonym	Hypernym	Hyponym
genetics	biology	genomics
genetic science	biological science	

the branch of biology that studies heredity and variation in organisms

Related terms to 'genetics'

term	POS	score
genetics	noun	5
gen	noun	4
amino acid	noun	4
gene	noun	3
protein	noun	3
chromosome	noun	3
dna	noun	2

Root terms

	term	POS
1	genetics	noun
2	gene	noun
3	amino acid	noun
4	protein	noun
5		
6		

Expanded terms

	label	POS	parent	relation	depth
1	genetics	noun	-	ROOT	0
2	gene	noun	-	ROOT	0
3	gen	noun	gene	SYNONYM	1
4	amino acid	noun	-	ROOT	0
5	alanine	noun	amino acid	HYPONYM	1
6	protein	noun	-	ROOT	0

Figure 2. WebCORE problem definition phase

a discussion on a mailing list about the system of organization developed in Delicious[5] and Flickr[6]. It is associated to those information retrieval methodologies consisting of collaboratively generated, open-ended labels that categorize content.

Although they suffer from problems of imprecision and ambiguity, techniques employing free-form tagging encourage users to organize information in their own ways and actively interact with the system.

3.2 Automatic Ontology Recommendation

Once the user has selected the most appropriate set of terms to describe the problem domain, the tool performs the processes of ontology retrieval and ranking. These processes play a key role within the system, since they provide the first level of information to the user. To enhance the previous approaches of CORE, an adaptation of traditional Information Retrieval techniques have been integrated into the system. Our novel strategy to ontology retrieval can be seen as an evolution of classic keyword-based retrieval techniques [21], where textual documents are replaced by ontologies.

3.2.1 Query encoding and ontology retrieval

The queries supported by our model are expressed using the terms selected during the Golden Standard definition phase.

In classic keyword-based vector-space models for information retrieval [21], each of the query keywords is assigned a weight that represents the importance of the keyword in the information need expressed by the query, or its discriminating power for discerning relevant from irrelevant documents.

Analogously, in our model, the terms included in the Golden Standard can be weighted to indicate the relative interest of the user for each of the terms to be explicitly mentioned in the ontologies. In our system, these weights are automatically assigned considering the depth measure of each of the terms included in the Golden Standard.

Let T be the set of all terms defined in the Golden Standard definition phase. Let d_i be the depth measure associate with each term $t_i \in T$. Let q be query vector extracted from the Golden Standard definition, and let w_i be the weight associated to each of these terms, where for each $t_i \in \mathrm{T}$, $w_i \in [0,1]$. Then, the weight w_i is calculated as:

$$w_i = \frac{1}{d_i + 1}$$

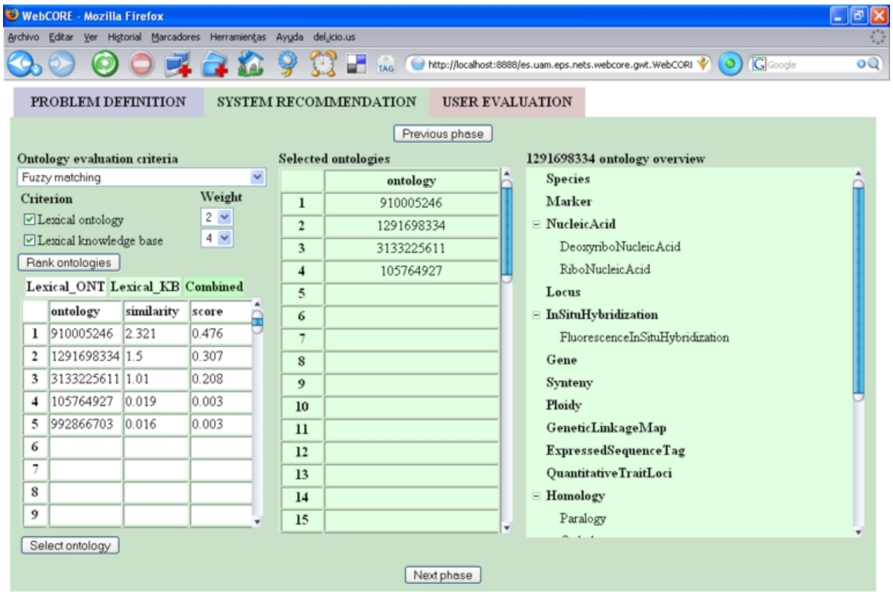
This measure gives more relevance to the terms explicitly expressed by the user, and less importance to those ones extended or derived from previously selected terms. An interesting future work could be to enhance and refine the query, e.g. based on terms popularity, or other more complex strategies as terms frequency analysis.

To carry out the process of ontology retrieval, the approach is focused on the lexical level, retrieving those ontologies that contain a subset of the terms expressed by the user during the Golden Standard definition. To compute the matching, two different options are available within the tool: search for exact matches and search for matches based on the Levenshtein distance between two terms.

In both cases, the query execution returns a set of ontologies that satisfy user requirements. Considering that not all the retrieved ontologies fulfil the same level of satisfaction, it is the system task to sort them and present the ranked list to the user.

[5] del.icio.us - social bookmarking, http://del.icio.us/

[6] Flickr - photo sharing, http://www.flickr.com/


WebCORE - Mozilla Firefox
Archivo Editar Ver Historial Marcadores Herramientas Ayuda del.icio.us
http://localhost:8888/es.uam.eps.nets.webcore.gwt.WebCORE
Google
PROBLEM DEFINITION
SYSTEM RECOMMENDATION
USER EVALUATION
Previous phase
Rank ontologies
Lexical_ONT
Lexical_KB
Combined
Select ontology
Next phase

Ontology evaluation criteria

Fuzzy matching

Criterion	Weight
☑ Lexical ontology	2
☑ Lexical knowledge base	4

	ontology	similarity	score
1	910005246	2.321	0.476
2	1291698334	1.5	0.307
3	3133225611	1.01	0.208
4	105764927	0.019	0.003
5	992866703	0.016	0.003
6			
7			
8			
9			

Selected ontologies

	ontology
1	910005246
2	1291698334
3	3133225611
4	105764927
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	

1291698334 ontology overview

- Species
- Marker
- NucleicAcid
  - DeoxyriboNucleicAcid
  - RiboNucleicAcid
- Locus
- InSituHybridization
  - FluorescenceInSituHybridization
- Gene
- Synteny
- Ploidy
- GeneticLinkageMap
- ExpressedSequenceTag
- QuantitativeTraitLoci
- Homology
  - Paralogy

Figure 3. WebCORE system recommendation phase

3.2.2 Ontology ranking

Once the list of ontologies is formed, the ontology-search engine computes a semantic similarity value between the query and each ontology as follows. We represent each ontology in the search space as an ontology vector $o_j \in O$, where o_{ji} is the mean of the term t_i similarities with all the matched entities in the ontology if any matching exists, and zero otherwise.

The components o_{ji} are calculated as:

$$o_{ji} = \left| M_{ji} \right| \frac{\sum_{M_{ji}} w(m_{ji})}{\sum_{M_i} w(m_i)}$$

where M_{ji} is the set of matches of the term t_i in the ontology o_j, $w(m_{ji})$ represents the similarities between the term t_i and the entities of the ontology o_j that matches with it, M_i is the set of matches of the term t_i within all the ontologies and $w(m_i)$ represents the weights of each of these matches.

For example, if we define in the Golden Standard a term “acid”, this term may return several matches in the same ontology with different entities as: “acid”, “amino acid”, etc. In order to establish the appropriate weight in the ontology vector, o_{ij}, the goal is to compute the number of matches of one term in the whole repository of ontologies and give more relevance to those ontologies that have matched that specific term more times.

Due to the way in which the vector o_j is constructed, each component o_{ij} contains specific information about the similarity between the ontology and the corresponding term t_i. To compute the final similarity between the query vector q and the ontology vector o_j, the vectorial model calculates the cosine measure between both vectors. However, if we follow the traditional vectorial model, we will only be considering the difference between the query and the ontology vectors according to the angle they form, but not taking into account their dimensions. Thus, to overcome this limitation, the above cosine measure used in the vectorial model has been replaced by the simple dot product. Hence, the similarity measure between an ontology o_j and the query q is simply compute as follows:

$$sim(q, o_j) = q \cdot o_j$$

3.2.3 Combination with Knowledge Base Retrieval

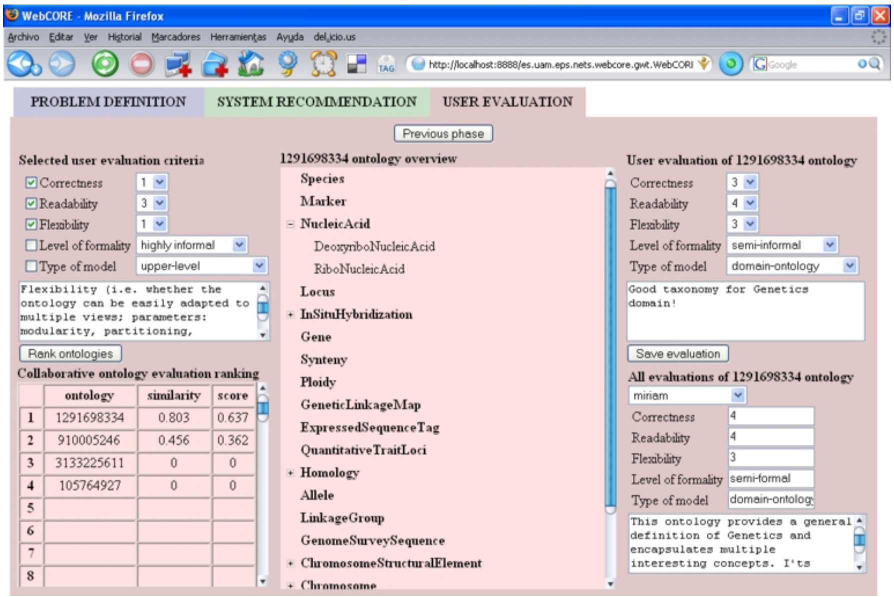
If the knowledge in the ontology is incomplete, the ontology ranking algorithm performs very poorly. Queries will return less results than expected, the relevant ontologies will not be retrieved, or will get a much lower similarity value than it should. For instance, if there are ontologies about “restaurants”, and “dishes” are expressed as instances in the corresponding Knowledge Base (KB), a user searching for ontologies in this domain may be also interested in the instances and literals contained in the KB. To cope with this issue, our ranking model combines the similarity obtained from the terms that belong to the ontology with the similarity obtained from the terms that belong to the KB using the adaptation of the vector space model explained before.

On the other hand, the combination of outputs of several search engines has been a widely addressed research topic in the Information Retrieval field [9]. After testing several approaches, we have selected the so-called Comb-MNZ strategy. This technique has been shown in prior works as one of the simplest and most effective rank aggregation techniques, and consists of computing a combined ranking score by a linear combination of the input scores with additional factors that measure the relevance of each score in the final ranking. In our case, the relevancies of the scores, i.e., the relevancies of the similarity computation within the ontology and within the knowledge base, are given by the user. He can select a value $v_i \in [1, 5]$ for each kind of search, and this value is then mapped to a corresponding value s_i using the following normalization.

$$s_i = \frac{v_i}{5}$$

Following this idea, the final score is computed as:

$$s_O \times sim(q, o) + s_{kb} \times sim(q, kb)$$


WebCORE - Mozilla Firefox
Archivo Editar Ver Historial Marcadores Herramientas Ayuda del.icio.us
http://localhost:8888/es.uam.eps.nets.webcore.gwt.WebCORE
Google
PROBLEM DEFINITION
SYSTEM RECOMMENDATION
USER EVALUATION
Previous phase
Rank ontologies
Save evaluation

Selected user evaluation criteria

- ☑ Correctness: 1
- ☑ Readability: 3
- ☑ Flexibility: 1
- ☐ Level of formality: highly informal
- ☐ Type of model: upper-level

```
Flexibility (i.e. whether the
ontology can be easily adapted to
multiple views; parameters:
modularity, partitioning,
```

Collaborative ontology evaluation ranking

	ontology	similarity	score
1	1291698334	0.803	0.637
2	910005246	0.456	0.362
3	3133225611	0	0
4	105764927	0	0
5			
6			
7			
8			

1291698334 ontology overview

- Species
- Marker
- ⊟ NucleicAcid
  - DeoxyriboNucleicAcid
  - RiboNucleicAcid
- Locus
- + InSituHybridization
- Gene
- Synteny
- Ploidy
- GeneticLinkageMap
- ExpressedSequenceTag
- QuantitativeTraitLoci
- + Homology
- Allele
- LinkageGroup
- GenomeSurveySequence
- + ChromosomeStructuralElement
- + Chromosome

User evaluation of 1291698334 ontology

- Correctness: 3
- Readability: 4
- Flexibility: 3
- Level of formality: semi-informal
- Type of model: domain-ontology

```
Good taxonomy for Genetics
domain!
```

All evaluations of 1291698334 ontology

miriam

- Correctness: 4
- Readability: 4
- Flexibility: 3
- Level of formality: semi-formal
- Type of model: domain-ontolog

```
This ontology provides a general
definition of Genetics and
encapsulates multiple
interesting concepts. I'ts
```

Figure 4. WebCORE user evaluation phase

For future work, we are considering to set s_i using statistical information about the knowledge contained in the ontologies, the knowledge contained in the KBs and the information requested by the user during the Golden Standard definition phase.

Figure 3 shows the system recommendation interface. At the left side the user can select the matching methodology (fuzzy or exact), the search spaces (ontology entities and knowledge base entities), and the weight or importance given to each of the previously selected search spaces. In the right part the user can visualize the ontology and navigate across it. Finally, the middle of the interface presents the list of ontologies selected for the user to be evaluated during the collaborative evaluation phase.

3.3 Collaborative Ontology Evaluation

The third and last phase of the system is compound of a novel ontology recommendation algorithm that exploits the advantages of Collaborative Filtering [1], exploring the manual evaluations stored in the system to rank the set of ontologies that best fulfils the user's interests.

In WebCORE, user evaluations are represented as a set of five different criteria [15] and their respective values, manually determined by the users who made the evaluations.

- **Correctness**: specifies whether the information stored in the ontology is true, independently of the domain of interest.
- **Readability**: indicates the non-ambiguous interpretation of the meaning of the concept names.
- **Flexibility**: points out the adaptability or capability of the ontology to change.
- **Level of formality**: highly informal, semi-informal, semi-formal, rigorously-formal.
- **Type of model**: upper-level (for ontologies describing general, domain-independent concepts), core-ontologies (for ontologies that contain the most important concepts on a specific domain), domain-ontologies (for ontologies that broadly describe a domain), task-ontologies (for ontologies focused on generic types of tasks or activities) and application-ontologies (for ontologies describing a domain in an application-dependent manner).

The above criteria can have discrete numeric or non-numeric values. The user's interests are expressed like a subset of these criteria, and their respective values, meaning thresholds or restrictions to be satisfied by user evaluations. Thus, a numeric criterion will be satisfied if an evaluation value is equal or greater than that expressed by its interest threshold, while a non-numeric criterion will be satisfied only when the evaluation is exactly the given threshold (i.e. in a Boolean or yes/no manner).

According to both types of user evaluation and interest criteria, numeric and Boolean, the recommendation algorithm will measure the degree in which each user restriction is satisfied by the evaluations, and will recommend a ranked ontology list according to similarity measures between the thresholds and the collaborative evaluations. To create the final ranked ontology list the recommender module follows two phases. In the first one it calculates the similarity degrees between all the user evaluations and the specified user interest criteria thresholds. In the second one it combines the similarity measures of the evaluations, generating the overall rankings of the ontologies.

Figure 4 shows all the previous definitions and ideas, locating them in the graphical interface of the system. On the left side of the screen, the user introduces the thresholds for the recommendations and obtains the final collaborative ontology ranking. On the right side, the user adds new evaluations for the ontologies and checks evaluations given by the rest of the users.

3.3.1 *Collaborative Evaluation Measures*

As mentioned before, a user evaluates an ontology considering five different criteria that can be divided in two different groups: a) numeric criteria ('correctness', 'readability' and 'flexibility'), which take discrete numeric values [1, 2, 3, 4, 5], where 1 means the ontology does not fulfil the criterion, and 5 means the ontology completely satisfies the criterion, and, b) Boolean criteria ('level of formality' and 'type of model'), which are represented by specific non-numeric values that can be or not satisfied by the ontology.

Taking into account the previous definitions, user interests will be a subset of the above criteria and their respective values representing the set of thresholds that should be reached by the ontologies. Given a set of user interests, the system will size up all the stored evaluations, and will calculate their similarity measures.

To explain these similarities we shall use a simple example of six different evaluations (E_1, E_2, E_3, E_4, E_5 and E_6) of a given ontology. In the explanation we shall distinguish between the numeric and the Boolean criteria. We start with the Boolean ones, assuming two different criteria, C_1 and C_2, with three possible values: "A", "B" and "C". In Table 1 we show the threshold values established by a user for these two criteria, "A" for C_1 and "B" for C_2, and the six evaluations stored in the system.

Table 2. Thresholds and evaluations for Boolean criteria C_1 and C_2

		Evaluations					
Criteria	Thresholds	E_1	E_2	E_3	E_4	E_5	E_6
C_1	"A"	"A"	"B"	"A"	"C"	"A"	"B"
C_2	"B"	"A"	"A"	"B"	"C"	"A"	"A"

In this case, because of the threshold of a criterion *n* is satisfied or not by a certain evaluation *m*, their corresponding similarity measure is simply 0 if they have the same value, and 2 otherwise.

$$similarity_{bool}(criterion_{mn}) = \begin{cases} 0 & \text{if } evaluation_{mn} \neq threshold_{mn} \\ 2 & \text{if } evaluation_{mn} = threshold_{mn} \end{cases}$$

The similarity results for the Boolean criteria of the example are shown in Table 3.

Table 3. Similarity values for Boolean criteria C_1 and C_2

		Evaluations					
Criteria	Thresholds	E_1	E_2	E_3	E_4	E_5	E_6
C_1	"A"	**2**	**0**	**2**	**0**	**2**	**0**
C_2	"B"	**0**	**0**	**2**	**0**	**0**	**0**

For the numeric criteria, the evaluations can overcome the thresholds to different degrees. Table 4 shows the thresholds established for criteria C_3, C_4 and C_5, and their six available evaluations. Note that E_1, E_2, E_3 and E_4 satisfy all the criteria, while E_5 and E_6 do not reach some of the corresponding thresholds.

Table 4. Thresholds and evaluations for numeric criteria C_3,C_4 and C_5

		Evaluations					
Criteria	Thresholds	E_1	E_2	E_3	E_4	E_5	E_6
C_3	≥ 3	3	4	5	5	2	0
C_4	≥ 0	0	1	4	5	0	0
C_5	≥ 5	5	5	5	5	4	0

In this case, the similarity measure has to take into account two different issues: the degree of satisfaction of the threshold, and the difficulty of achieving its value. Thus, the similarity between the value of criterion *n* in the evaluation *m*, and the threshold of interest is divided into two factors: 1) a similarity factor that considers whether the threshold is surpassed or not, and, 2) a penalty factor which penalizes those thresholds that are easier to be satisfied.

$$similarity_{num}(criterion_{mn}) =$$
$$= 1 + similarity^{*}_{num}(criterion_{mn}) \cdot penalty_{num}(threshold_{mn}) \in [0,2]$$

This measure will also return values between 0 and 2. The idea of returning a similarity value between 0 and 2 is inspired on other collaborative matching measures [18] to not manage negative numbers, and facilitate, as we shall show in the next subsection, a coherent calculation of the final ontology rankings.

The similarity assessment is based on the distance between the value of the criterion *n* in the evaluation *m*, and the threshold indicated in the user's interests for that criterion. The more the value of the criterion *n* in evaluation *m* overcomes the threshold, the greater the similarity value shall be.

Specifically, following the expression below, if the difference *dif = (evaluation – threshold)* is equal or greater than 0, we assign a positive similarity in (0,1] that depends on the maximum difference *maxDif = (maxValue – threshold)* we can achieve with the given threshold; and else, if the difference *dif* is lower than 0, we give a negative similarity in [-1,0), punishing the distance of the value with the threshold.

$$similarity^{*}_{num}(criterion_{mn}) = \begin{cases} \dfrac{1+dif}{1+maxDif} \in (0,1] & \text{if } dif \geq 0 \\ \dfrac{dif}{threshold} \in [-1,0) & \text{if } dif < 0 \end{cases}$$

Table 5 summarizes the similarity* values for the three numeric criteria and the six evaluations of the example.

Table 5. Similarity* values for numeric criteria C_3, C_4 and C_5

		Evaluations					
Criteria	Thresholds	E_1	E_2	E_3	E_4	E_5	E_6
C_3	≥ 3	1/4	2/4	3/4	3/4	-1/3	-1
C_4	≥ 0	1/6	2/6	5/6	1	1/6	1/6
C_5	≥ 5	1	1	1	1	-1/5	-1

Comparing the evaluation values of Table 4 with the similarity values of Table 5, the reader may notice several important facts:

1. Evaluation E_4 satisfies criteria C_4 and C_5 with evaluations of 5. Applying the above expression, these criteria receive the same similarity of 1. However, criterion C_4 has a threshold of 0, and C_5 has a threshold equal to 5. As it is more difficult to satisfy the restriction imposed to C_5, this one should have a greater influence in the final ranking.
2. Evaluation E_6 gives an evaluation of 0 to criteria C_3 and C_5, not satisfying either of them and generating the same similarity value of -1. Again, because of their different thresholds, we should distinguish their corresponding relevance degrees in the rankings.

For these reasons, a threshold penalty is applied, reflecting how difficult it is to overcome the given thresholds. The more difficult to surpass a threshold, the lower the penalty value shall be.

$$penalty_{num}(threshold) = \frac{1+threshold}{1+maxValue} \in (0,1]$$

Table 6 shows the threshold penalty values for the three numeric criteria and the six evaluations of the example.

Table 6. Threshold penalty values for numeric criteria C_3, C_4 and C_5

		Evaluations					
Criteria	Thresholds	E_1	E_2	E_3	E_4	E_5	E_6
C_3	≥ 3	4/6	4/6	4/6	4/6	4/6	4/6
C_4	≥ 0	1/6	1/6	1/6	1/6	1/6	1/6
C_5	≥ 5	1	1	1	1	1	1

Finally, the similarity results for the numeric criteria of the example are shown in Table 7.

Table 7. Similarity values for numeric criteria C_3, C_4 and C_5

		Evaluations					
Criteria	Thresholds	E_1	E_2	E_3	E_4	E_5	E_6
C_3	≥ 3	**1.17**	**1.33**	**1.5**	**1.5**	**0.78**	**0.33**
C_4	≥ 0	**1.03**	**1.05**	**1.14**	**1.17**	**1.03**	**1.03**
C_5	≥ 5	**2**	**2**	**2**	**2**	**0.5**	**0**

As a preliminary approach, we calculate the similarity between an ontology evaluation and the user's requirements as the average of its *N* criteria similarities.

$$similarity(evaluation_m) = \frac{1}{N}\sum_{n=1}^{N} similarity(criterion_{mn})$$

A weighted average could be even more appropriate, and might make the collaborative recommender module more sophisticated and adjustable to user needs. This will be considered for a possible enhancement of the system in the continuation of our research.

3.3.2 Collaborative Ontology Ranking

Once the similarities are calculated taking into account the user's interests and the evaluations stored in the system, a ranking is assigned to the ontologies.

The ranking of a specific ontology is measured as the average of its *M* evaluation similarities. Again, we do not consider different priorities in the evaluations of several users. We have planned to include in the system personalized user appreciations about the opinions of the rest of the users. Thus, for a certain user some evaluations will have more relevance than others, according to the users that made it.

$$\begin{aligned} ranking(ontology) &= \frac{1}{M}\sum_{m=1}^{M} similarity(evaluation_m) \\ &= \frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N} similarity(criterion_{mn}) \end{aligned}$$

Finally, in case of ties, the collaborative ranking mechanism sorts the ontologies taking into account not only the average similarity between the ontologies and the evaluations stored in the system, but also the total number of evaluations of each ontology, providing thus more relevance to those ontologies that have been rated more times.

$$\frac{M}{M_{total}} ranking(ontology)$$

4. EXPERIMENTS

In this section, we present some early experiments that attempt to measure: a) the gain of efficiency and effectiveness, and the b) increment of users' satisfaction obtained with the use of our system when searching ontologies within a specific domain.

The scenario of the experiments was the following. A repository of thirty ontologies was considered and eighteen subjects participated in the evaluations. They were Computer Science Ph.D. students of our department, all of them with some expertise in modeling and exploitation of ontologies. They were asked to search and evaluate ontologies with WebCORE in three different tasks. For each task and each student, one of the following problem domains was selected:

- **Family**. Search for ontologies including family members: mother, father, daughter, son, etc.
- **Genetics**. Search for ontologies containing specific vocabulary of Genetics: genes, proteins, amino acids, etc.
- **Restaurant**. Search for ontologies with vocabulary related to restaurants: food, drinks, waiters, etc.

In the repository, there were six different ontologies related to each of the above domains, and twelve ontologies describing other no related knowledge areas. No information about the domains and the existent ontologies was given to the students.

Tasks 1 and 2 were performed first without the help of the collaborative modules of the system, i.e., the term recommender of the problem definition phase and the collaborative ranking of the user evaluation phase. After all users finished the previous ontology searches and evaluations, task 3 was done with the collaborative components activated. For each task and each student, we measured the time expended, and the number of ontologies retrieved and selected ('reused'). We also asked the users about their satisfaction (in a 1-5 rating scale) about each of the selected ontologies and the collaborative modules.

Tables 8 and 9 contain a summary of the obtained results. Note that measures of task 1 are not shown. We have decided not to consider them for evaluation purposes because we discern the first task as a learning process of the use of the tool, and its time executions and number of selected ontologies as skewed no objective measures.

To evaluate the enhancements in terms of efficiency and effectiveness, we present in Table 8 the average number of reused ontologies and the average execution times for task 2 and 3. The results show a significant improvement when the collaborative modules of the system were activated. In all the cases, the students made use of the terms and evaluations suggested by others, accelerating the processes of problem definition and relevant ontology retrieval.

Table 8. Average number of reused ontologies and execution times (in minutes) for tasks 2 and 3

	Task 2 (without collaborative modules)	**Task 3** (with collaborative modules)	% improvement
# reused ontologies	3.45	4.35	26.08
execution time	9.3	7.1	23.8

On the other hand, table 9 shows the average degrees of satisfaction revealed by the users about the retrieved ontologies and the collaborative modules. Again, the results evidence positive applications of our approach.

Table 9. Average satisfactions values (1-5 rating scale) for ontologies reused in tasks 2 and 3, collaborative recommendations and rankings

Task 2	**Task 3**	% improvement	Initial term recommendation	Final ontology ranking
3.34	3.56	6.58	4.7	4.4

5. CONCLUSIONS AND FUTURE WORK

In this paper, a web application for ontology evaluation and reuse has been presented. The novel aspects of our proposal include the use of WordNet to help users to define the Golden Standard; a new ontology retrieval technique based on traditional Information Retrieval models; rank fusion techniques to combine different ontology evaluation measures; and two collaborative modules: one that suggests the most popular terms for a given domain, and one that recommends lists of ontologies with a multi-criteria strategy that takes into account user opinions about ontology features that can only be assessed by humans.

6. ACKNOWLEDGMENTS

This research was supported by the Spanish Ministry of Science and Education (TIN2005-06885 and FPU program).

7. REFERENCES

[1] Adomavicius, G., and Tuzhilin, A.: *Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions*. IEEE Transactions on Knowledge and Data Engineering 17(6): 734-749, 2005.

[2] Alani, H., and Brewster, C.: *Metrics for Ranking Ontologies*. Proceedings of the 4th Int. Workshop on Evaluation of Ontologies for the Web (EON'06), at the 15th Int. World Wide Web Conference (WWW'06). Edinburgh, UK, 2006.

[3] Alani, H., Brewster, C., and Shadbolt, N.: *Ranking Ontologies with AKTiveRank*. Proc.. of the 5th Int. Semantic Web Conference (ISWC'06). Athens, Georgia, USA, 2006.

[4] Brank J., Grobelnik M., and Mladenic D.: *A Survey of Ontology Evaluation Techniques*. Proceedings of the 4th Conference on Data Mining and Data Warehouses (SiKDD'05), at the 7th Int. Multi-conference on Information Society (IS'05). Ljubljana, Slovenia, 2005.

[5] Brewster, C., Alani, H., Dasmahapatra, S. and Wilks, Y. *Data driven ontology evaluation*. Proc. of the 4th Int. Conf. on Language Resources and Evaluation (LREC04). Lisbon 2004

[6] Ding, Y., and Fensel, D.: *Ontology Library Systems: The key to successful Ontology Reuse*. Proc. of the 1st Semantic Web Working Symposium (SWWS'01). Stanford, CA, USA, 2001.

[7] Farquhar, A., Fikes, R., and Rice, J.: *The Ontolingua server: A tool for collaborative ontology construction*. Technical report, Stanford KSL 96-26, 1996.

[8] Fernández, M., Cantador, I., and Castells, P. *CORE: A Tool for Collaborative Ontology Reuse and Evaluation*. Proceedings of the 4th Int. Workshop on Evaluation of Ontologies for the Web (EON'06), at the 15th Int. World Wide Web Conference (WWW'06). Edinburgh, UK, 2006.

[9] Lee, J. H.: *Analysis of multiple evidence combination*. Proceedings of the 20th ACM Int. Conference on Research and Development in IR (SIGIR'97). New York, 1997.

[10] Lozano-Tello, A., and Gómez-Pérez, A.: *Ontometric: A method to choose the appropriate ontology*. Journal of Database Management, 15(2):1–18, 2004.

[11] Maedche, A., and Staab, S.: *Measuring similarity between ontologies*. Proceedings of the 13th European Conference on Knowledge Acquisition and Management (EKAW 2002). Madrid, Spain, 2002.

[12] Miller, G. A.: *WordNet: A lexical database for English. New horizons in commercial and industrial Artificial Intelligence*. Communications of the Association for Computing Machinery, 38(11): 39-41, 1995.

[13] Montaner, M., López, B., and De la Rosa, J.L.: *A Taxonomy of Recommended Agents on the Internet*. Artificial intelligence Review 19: 285-330, 2003.

[14] Noy, N. F., Chugh, A., Liu, W., and Musen, M. A.: *A Framework for Ontology Evolution in Collaborative Environments*. Proceedings of the 5th Int. Semantic Web Conference (ISWC'06). Athens, Georgia, USA, 2006.

[15] Paslaru, E.: *Using Context Information to Improve Ontology Reuse*. Doctoral Workshop at the 17th Conference on Advanced Information Systems Engineering (CAiSE'05). Porto, Portugal, 2005.

[16] Porzel, R., and Malaka, R.: *A task-based approach for ontology evaluation*. Proc. of the 16th European Conference on Artificial Intelligence (ECAI'04). Valencia, Spain, 2004.

[17] *Protégé OWL ontology Repository*. http://protege.stanford.edu/download/ontologies.html

[18] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J.: *GroupLens: An Open Architecture for Collaborative Filtering of Netnews*. Internal Research Report, MIT Center for Coordination Science, 1994.

[19] Sabou, M., López, V., Motta, E., and Uren, V.: *Ontology Evaluation on the Real Semantic Web*. Proceedings of the 4th Int. Workshop on Evaluation of Ontologies for the Web (EON'06), at the 15th Int. World Wide Web Conference (WWW'06). Edinburgh, UK, 2006.

[20] Sabou, M., López, V., Motta, E., and Uren, V.: *Ontology Selection for the Real Semantic Web: How to cover the Queen's Birthday Dinner?* Proc. of the 15th International Conference on Knowledge Engineering and Knowledge Management (EKAW'06). Podebrady, Czech Republic, 2006.

[21] Salton, G., and McGill, M.: *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, 1983.

[22] Smith, G.: *Atomiq: Folksonomy: Social Classification*. 2004. http://atomiq.org/archives/2004/08/folksonomy_social_classification.html

[23] Sure, Y., Erdmann, M., Angele, J., Staab, S., Studer, R., and Wenke, D.: *OntoEdit: Collaborative Ontology Development for the Semantic Web*. Proceedings of the 1st International Semantic Web Conference (ISWC '02), Sardinia, Italy, 2002.

[24] *Swoogle - Semantic Web Search Engine*. http://swoogle.umbc.edu