An Ontology-driven Document Retrieval Strategy for Organisational Knowledge Management Systems

Carlos M. Toledo[1], Mariel A. Ale[2], Omar Chiotti[1], and María R. Galli[1]

[1] INGAR-CONICET
Institute of Development and Design
National Council of Scientific and Technological Research
Avellaneda 3657, Santa Fe, Argentina
{cmtoledo,mrgalli,chiotti}@santafe-conicet.gov.ar,
http://www.ingar.santafe-conicet.gov.ar/
[2] CIDISI - UTN
Research and Development Center of Information System Engineering
National Technological University
Lavaisse 610, Santa Fe, Argentina
male@utn.frsf.edu.ar
http://cidisi.frsf.utn.edu.ar/

Abstract. Enterprises are inserted in a competitive environment in which knowledge is vital to survive in the current global market. Competition is no longer conceived as it was in traditional markets. In this global market, knowledge is considered an asset that has an economic value for an organization and a strategic resource used to increase productivity and offer stability in dynamic competitive environments. Such significance of knowledge implies the need for protecting this vital resource by safeguarding the right access, its persistence over time, and its adequate retrieval. In this work, we propose an organizational memory architecture, and annotation and retrieval information strategies based on domain ontologies that take in account complex words to retrieve information through natural language queries. To test these strategies, we implemented a flexible framework to experiment with knowledge retrieval approaches. Finally, experimental results are evaluated and analyzed through standard measures.

Keywords: Document annotation, Knowledge management framework, Domain ontology, Knowledge retrieval

1 Introduction

Enterprises are inserted in a competitive environment in which knowledge is vital to survive in the current global market. Competition is no longer conceived as it was in traditional markets. In this global market, knowledge is considered an asset that has an economic value for an organization and a strategic resource used to increase productivity and offer stability in dynamic competitive environments [13]. This importance is further increased in enterprises where the development process requires a high degree of intellectual capital.

In this new paradigm, in which physical capital and work are no longer the only fundamental bases for successful management, Knowledge Management (KM) has captured the attention of enterprises as one of the most promising ways to reach success in this information era [23]. Companies are beginning to understand the importance of knowledge in the organization as a resource which enables them to obtain a sustainable competitive advantage [6].

Such significance given to knowledge implies the necessity to ensure the protection of this vital resource by means of safeguarding the right access, its persistence over time, and its adequate retrieval. KM aims at solving problems related to knowledge identification, creation, codification, storage, diffusion, and access to promote learning and innovation [25]; and it provides an infrastructure to bring the right knowledge to the right people in the right form and at the right time [27].

There are many KM initiatives implemented in organisations, but most of these efforts often fail to manage the natural heterogeneity of organisational knowledge sources, and present no adaptation capabilities to add new knowledge sources. For this reason, a system that allows each organizational area to autonomously administrate its own knowledge repository (Knowledge autonomy principle), and provides the means to share this knowledge with other organizational areas (Coordination principle) is an adequate solution for managing complex organizational knowledge sources [28].

Ale et al. [2, 3] propose a set of social, technological, cultural, political, and economical requirements that a KM model should fulfill. In order to accomplish these requirements, they present a novel KM conceptual model that involves both social and technological perspectives. This model is a framework for developing organizational memories that capture, increase, store, organize, analyze, and share organizational knowledge. From the technological perspective, they describe an ontology-driven KM architecture called Onto-DOM.

Onto-DOM uses an ontology approach to annotate and retrieve information based on a distributed organizational memory strategy that processes users' natural language queries (Figure 1). Domain ontologies aim at capturing domain knowledge, providing a compromised understanding of a domain, and allowing integration between different formats of knowledge and information sources [16]. This strategy selects ontological concepts derived from the nouns in a document as document descriptors. Therefore, it selects only domain relevant concepts and provides a homogeneous representation of structurally heterogeneous objects: documents.

The document annotation and retrieval algorithms used in Onto-DOM do not consider complex concepts composed of adjectives or more that one noun, nor named entities (name of cities, persons, localizations, etc.). When the ontological engineer annotates a document, it selects the document nouns or synonyms of these nouns that coincide with ontological concepts. Other syntactical categories (adjectives, pronouns, etc.) are ruled out. Although nouns frequently

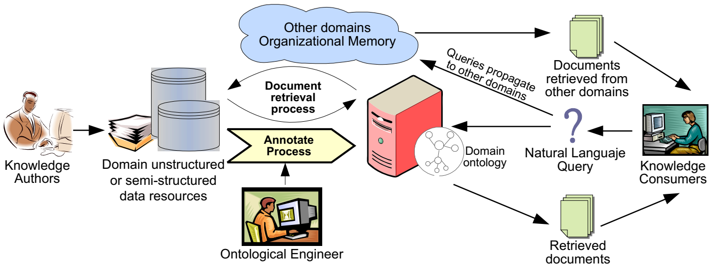
Other domains Organizational Memory
Documents retrieved from other domains
Queries propagate to other domains
Document retrieval process
Annotate Process
Knowledge Authors
Domain unstructured or semi-structured data resources
Ontological Engineer
Domain ontology
?
Natural Languaje Query
Knowledge Consumers
Retrieved documents

Fig. 1. Onto-DOM knowledge annotation and retrieval processes

carry more semantics than adjectives, adverbs, or verbs [5], the modifiers of a word can completely change its meaning. For example, in an enterprise that sells furniture, *back formal dinning table* and *mid-century danish table* are different concepts. The whole semantics of a document should be kept by storing all its representative document concepts, even the complex ones.

We propose an extension to document annotation and retrieval strategies used by Onto-DOM. These strategies take in account the modifiers of nouns (e.g. adjectives) keeping all the semantics of a document. The rest of this work presents an integration among tools, techniques, and ontology-driven approaches for knowledge retrieval and document semantic annotation. The rest of the paper is organized as follows: Section 2 revises related work. Section 3 presents a KM system based on a distributed organizational memory [1]. Sections 4 and 5 describe strategies for document annotation and knowledge retrieval that use complex nouns and natural language queries. In Section 6 these strategies are evaluated. Finally, we present the conclusions.

2 Related work

Since beginning of semantic annotation, several work has been done. The first works in this area come from the Semantic Web [26], but recently, with the advent of organizational knowledge management, many frameworks and tools have arisen. A wide review of the state of the art of semantic annotation for KM is made in [31]. This work describes languages, tools, and requirements that should be taken in account to make sematic annotation systems for KM.

In the semantic web area, SHOE [22] system and Ontobroker [15] were the first attempts to enable semantic annotation of web documents, allowing web page authors to manually annotate their documents with machine-readable metadata. Nevertheless, manual annotation is an expensive process and it often leads to a knowledge acquisition bottleneck. To overcome this problem, semi-automatic annotation of documents has been proposed.

Knowledge Information Management (KIM) platform [20] uses the Sesame RDF [8] for ontology and knowledge base storage. The information extraction component of semantic annotation is performed by using components of GATE[3] tool kit [10]. SemTag [11] is the semantic annotation component of a comprehensive platform called Seeker, which performs large-scale annotations of web pages. In KIM, as well as in SemTag, annotation is considered the process of linking semantic descriptions to document entities.

Cerno [21] is a framework for semi-automatic semantic annotation of textual data using light-wight analysis techniques. It applies a method, based on software code analysis techniques, for semantic annotations. Cerno divides the text into constituents, makes a parse tree that contains natural language document fragments (e.g. paragraph function, expression, word, etc.), and identifies e-mails addresses, phone numbers, etc. Then, the process recognizes concept instances based on an annotation scheme that includes a list of concept names and domain vocabulary. Finally, the annotated text is stored in an external database. Since Cerno does not use ontologies, the relationships between concepts are not considered, which loosing relevant semantics.

Gschwandtner et. al. [17] propose a semantic annotation system to annotate free medical text. This approach uses MMTx [4] which maps concepts from the Unified Medical Language System (UMLS) metathesaurus [4] to concepts in the text. It also provides an interactive editor that facilitates the annotation of documents by MMTx, making it possible to create, visualize, and edit the semantic annotation of medical documents. This approach, unlike our own architecture, is domain dependent.

Ont-O-Mat [18] is an implementation of the S-CREAM semantic annotation framework [19]. The information extraction component is based on Amilcare *WordNet*[4]. Amilcare requires a training corpus of manually annotated documents. The approach uses *GATE* and *WordNet*[5] for natural language processing, and a domain ontology to provide the necessary context for knowledge objects. In our work, however, strategies do not have a learning phase, since only a replace of the domain ontology is needed to change the knowledge domain. We believe that these characteristics make this strategy suitable for a distributed organizational memory implementation, in which a large numbers of knowledge domains could appear.

Some other works that are worth mentioning are: PANKOW (Pattern-based ANnotation through Knowledge On the Web) [9], MnM [32], and Armadillo [12].

3 Conceptual architecture

In this work, an organizational memory architecture is presented. In this architecture, organizations are composed of several functional units (areas, departments, groups, practice communities, etc.). Each functional unit is a knowledge domain that fulfills autonomy and coordination principles allowing a KM approach based on a distributed architecture. Autonomy capabilities enable each

[3] General Architecture of Text Engineering - http://gate.ac.uk/

[4] Project web site: http://www.aktors.org/technologies/amilcare/

[5] Lexical database of English - http://wordnet.princeton.edu/

domain to manage local information, providing the possibility of choosing more appropriate perspectives, mechanisms and policy (e.g. security policy) to denote local knowledge. Coordination capabilities, on the other hand, enable each domain to exchange information with other domains, sharing knowledge between organizational units [7]. Each knowledge domain implements its own ontology-driven KM system (Figure 2).

The KM system of each domain has two main modules: *knowledge retrieval* and *knowledge representation* module. Figure 2 shows the proposed architecture with its interfaces. The knowledge representation module is responsible for retrieving knowledge from heterogeneous information sources (text documents, memorandum, spreadsheets, graphs, databases, etc.) and storing it in a suitable form. Knowledge acquisition and representation are crucial for an organizational memory system.

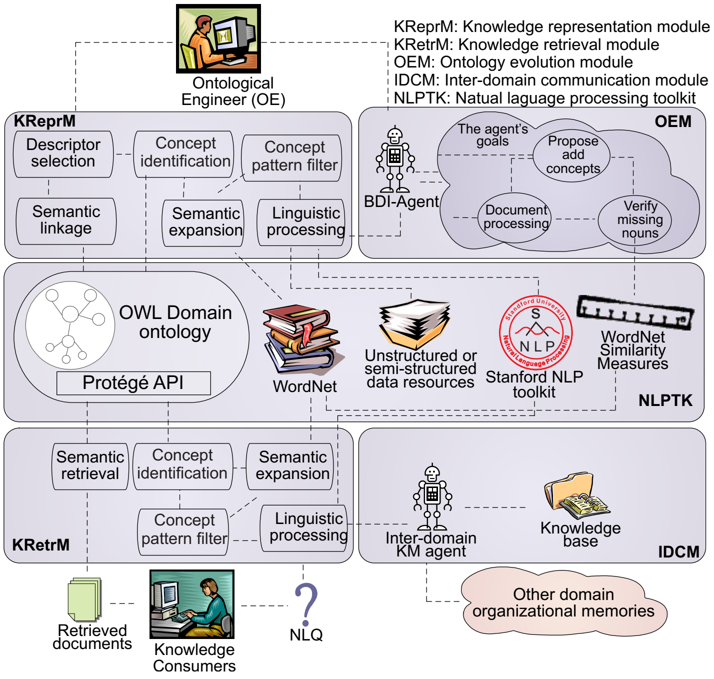

Fig. 2. The proposed conceptual architecture for the KM system

In the knowledge representation module [3], the ontological engineer stores the relationships between domain concepts, represented by ontological concepts in the domain ontology, and all unstructured and semi-structured data resources, so that the system can retrieve this information and provide the user with the correct document that carries the answer to a query. In the knowledge retrieval module [3], when a user makes a query in natural language, the system transforms this query by eliminating natural language ambiguity and then, it acquires the most representative concepts within the domain.

Additionally, the architecture has two modules: the *inter-domain communication* and the *ontology evolution*. The inter-domain communication module is responsible for propagating a user's query to another domain, in case this query cannot have a suitable response in the local domain. The ontology evolution module is responsible for keeping the domain ontology updated. Strategies used in these modules are out of the scope of this paper. For more details, you can see [1, 29, 30].

Knowledge storage and retrieval strategies implemented in this architecture are driven by domain ontologies. These ontologies capture the domain knowledge in a generic way, and provide a commonly agreed understanding of a domain, which can be reused, shared, and applied by groups and applications. Ontological concepts are used as the core of annotation and retrieval strategies. They enable the query refinement and reasoning processes (a process of generalization/specialization using ontology classes and subclasses). An additional benefit offered by ontologies and exploited in this architecture is context representation. It provides a domain model that allows storing the context into knowledge objects, facilitating a later reusability and interpretation.

4 Document annotation strategy

In an ideal case, an organizational memory information system would be self-adaptive and self-organized, collecting relevant knowledge during the operations of habitual business processes. Although this would diminish acquisition costs, it is not always possible an it often require some kind of manual intervention. In this strategy, it is the ontological engineer who performs this intervention. The ontological engineer has three main responsibilities: deciding which are the most representative domain concepts for each information source (so called document descriptors) in the document annotation strategy, developing the initial domain ontologies, and updating that ontology.

In the document annotation strategy implemented in the knowledge representation module, each document of the corpus goes through a linguistic analysis process (tokenization, lemmatization, and Part-Of-Speech Tagging or POSTagging)[14]. At the end of this process, a tag with its syntactic nature (adjective, singular noun, verb, present time, third person singular, conjunction, etc.) is assigned to each element or token (words and symbols) in the document. Once the tagging process is finished, document concepts (DC) are selected. The DCs selection is based on the following linguistic pattern: $DC = \{Noun^+ \mid (Adjetive^+ Noun^+\}$[6].

Using the selected DCs of the tagged document content, a searching process of descriptors begins. In this process, DCs are treated in the following way:

[6] Plus (+) sign represents one o more times

First, the process searches DC within the instances of concepts of the domain ontology as follows:

- The DC is compared with the instances of ontological concepts that represent named entities. If some coincidence is found, that DC is marked as a *candidate document descriptor* and it is linked to the concept instance.
- If there is no coincidence, the DC is added to an *unlinked DC list* for further treatment.

Second, the strategy searches for exact occurrences between DCs and the ontological concepts:

- With the unlinked DC list, each DC is compared with the ontological concepts. If some coincidence is found, that DC is marked as a candidate document descriptor, and it is linked to the concept. Such DC is eliminated from the unlinked DC list.
- If there is no coincidence, the head noun of such DC is compared with the ontological concepts. If some coincidence is found, that noun is marked as a candidate document descriptor, and it is linked to the concept. The DC is eliminated from the unlinked DC list.

Third, a process of semantic expansion extends the document representation as follows:

- For each unlinked DC , a set of synonyms of the head nouns of DCs are searched in WordNet. The head nouns of DCs are replaced by these synonyms, and the obtained DCs are compared with concepts. If some coincidence between a replaced DC and a concept is found, the original DC is marked as a candidate document descriptor. It is linked to the concept, and the original DC is eliminated from the unlinked DC list.

Finally, from the DCs that remain in the unlinked DC list, their hyperonyms are obtained by using WordNet, and the following process is developed:

- For each unlinked DC , a set of hyperonyms of its head nouns is searched in WordNet. These hyperonyms are replaced by the head nouns of DCs and, the obtained DCs are compared with the concepts. If some coincidence between an obtained DC and an concept is found, the original DC is marked as a candidate document descriptor. It is linked to the concept, and the original DC is eliminated from the unlinked DC list.

5 Knowledge retrieval strategy

When a user searches for information, it makes a natural language query. This query goes through a linguistic analysis process. Once query tokens have been labeled, nouns are selected and the following steps are executed:

Query concepts QCs are selected based on the same linguistic pattern used to select DCs ; then, QCs go through the following process:
Stage 1: direct coincidences between a QC and an instance of ontological concepts:

- Each *QC* is compared with the instances of ontological concepts that represent the knowledge in the domain. If a coincidence is found, this instance is selected and the ontological concept too.
- If there is no coincidence, the *QC* is added to an *unselected QC list* for further treatment.

Stage 2: direct coincidences between a *QC* and ontological concepts:

- With the unselected *QC* list, each *QC* is compared with the ontological concepts. If a coincidence is found, that concept is selected, and the *QC* is eliminated from the unselected *QC* list.
- If there is no coincidence, the head noun of such *QC* is compared with the ontological concepts. If a coincidence is found, that concept is selected, and the *QC* is eliminated from the unselected *QC* list.

Stage 3: coincidences between *QC* synonyms and hyperonyms of ontological concepts:

- For each unselected *QC* , a set of synonyms of the head noun of *QCs* is searched in WordNet. These synonyms are replaced by the head noun of *QCs* , and the obtained *QCs* are compared with the concepts. If a coincidence between a replaced *QC* and a concept is found, the concept is selected, and the original *QC* is eliminated from the unselected *QC* list.
- If there is no coincidence, the same process is executed, but the head noun of *QCs* is replaced by concepts hyperonyms.

Finally, the strategy selects only instances and concepts that provide an answer to the wh-word of the query (who: a person, where: a place, etc), and then, the documents associated with these instances and concepts are retrieved. The retrieved documents are sorted out so that the first documents have more precise answers to the query. This order is based on the percentage of descriptors that have coincided, and also, if these coincidences were with instances (more important descriptors), concepts, synonyms, or hyperonyms (less important descriptors).

6 Experimental results

The architecture has been implemented using Java language. Natural language processing is performed by means of NLP toolkits of the Stanford NLP Group[7]. The OWL ontology is accessed using Protégé-OWL API[8]. The WordNet lexical database is consulted by JAWS API[9].

A tourism enterprise was considered as a study case, which includes a knowledge domain: africa travel. An extension and specialization of the travel ontology[10], including specific terms employed in these domains, were used. This

[7] http://nlp.stanford.edu/

[8] http://protege.stanford.edu/

[9] http://lyle.smu.edu/ tspell/jaws/index.html

[10] Available at
http://protege.cim3.net/file/pub/ontologies/travel/travel.owl

ontology is formed by 267 concepts related to the vocabulary used in the domain. The corpus is composed of 125 documents, randomly selected from the Internet, with an average of 9187 words per document.

In this section, we evaluate the knowledge retrieval layer. For this reason, 50 questions are made. However, not all of the questions are shown in this paper due to space reasons. For each processed question, each retrieved document was classified as follows: contain an answer (C) or not contain an answer to the query (N). We considered that a document in correct if it directly or indirectly responds the question made. Since a bounded corpus with a well-known amount of documents is employed, the number of possible retrieval documents is limited, so we can know exactly what documents should be retrieved by a simple analysis of the corpus. This characteristic facilitates the analysis of result and allows using simple measures.

The following questions are used to analyze the proposed strategies:

q_j	Query
Q_1	What city offers Unesco world heritage excursions?
Q_2	What beach can I take a golf course?
Q_3	What hotel has a panoramic view?
Q_4	What city has a nightclub?
Q_5	What island offers water sport activities?
Q_6	Where can I play tennis?
Q_7	Where can I do yoga?
Q_8	What restaurant can I eat?
Q_9	Where can I go on a safari?
Q_{10}	What city has famous gardens?
Q_{11}	Where can I visit Kunene River?
Q_{12}	Where can I see elephants?
Q_{13}	What animals are there in the national park in Africa?
Q_{14}	What rivers are there in Africa?
Q_{15}	Where can I visit an old museum?
Q_{16}	Where is the natural history museum?
Q_{17}	Where can I do water sports?
Q_{18}	Where can I go deep sea fishing?
Q_{19}	Where can I do bird watching?
Q_{20}	Where can I watch lions?
...	...

For each query, a semantic analysis of the corpus document was made with the purpose of determining what documents that should be retried for such queries. The following table shows the results of the strategy for each query.

Query	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}	d_{11}	d_{12}	d_{13}	d_{14}	M_j
q_1	C	C	N	C	C	C	C								
q_2	C	C	C	C	C	C	C	C	C	C	C	C			
q_3	C	C	C	C	C	C	N	N	C						
q_4	C														
q_5	C														
q_6	C														
q_7	C	N	N	N	N	C	C								
q_8	C	C	N	C											
q_9	C	N	C												
q_{10}	C	C													
q_{11}	C	C	C												
q_{12}	C	C	N	C	C										
q_{13}	C	C	C	C	C	C	C	C	C	C					
q_{14}	C	C	C	C	C	C	C	C	C	C	C	C	C		
q_{15}	C	C	C	C	C	C	C	C	C						
q_{16}	C	C	N	N	C	N	C	C	N	C	C	N	N	C	
q_{17}	C	C	C												
q_{18}	C	C	C	C	C	C	C	C	C						
q_{19}	C	C	N	N	N	N	C	C	C						
q_{20}	C	C	C	C											
...	...	...	...	...	...	...	...	...	...	...	...	...			

q_j: query number j; d_k: document retrieved in the position k
C: Contain an answer; N: Non-contain an answer

In order to evaluate results quantitatively, the Mean Average Precision (MAP) measure is used. MAP is a standard measure among the TREC (Text REtrieval Conference) community[11], which determines the relation between the recall and precision measures of results obtained by the proposed knowledge retrieval algorithm.

Recall is the ratio of correctly found concepts (true positives) over the total number of representative concepts of documents (true positives and false negatives). Recall is meant to measure the degree of completeness of the strategy when retrieving all domain-relevant concepts [24].

$$r = \frac{tp}{tp+fn}$$

Precision is the ratio of "correctly" selected concepts by the strategy (true positives) over the total number of selected concepts by the strategy (true positives and false positives). This measure determines the strategy capability of not retrieving irrelevant concepts [24].

$$p = \frac{tp}{tp+fp}$$

MAP is the average of the precision value obtained from the set of top k relevant documents retrieved and this value is averaged over information needs.

$$MAP(Q) = \frac{1}{|Q|}\sum_{j=1}^{|Q|}\frac{1}{m_j}\sum_{k=1}^{m_j} Precision(R_{jk})$$

where, $q_j \in Q$ is the j query, $Q = \{q_1, q_2, ..., q_n\}$ is a set of query, $|Q|$ number of elements of the set Q (the number of queries), m_j is the number of the retrieval results to the query q_j, d_k is the retrieved document in the ranking k, R_{jk} is the set of ranked retrieval result from the top result until you get to document d_k, $R_{jk} = \{d_1, d_2, ..., d_k\}$

[11] Web site: http://trec.nist.gov/

The MAP formula calculates the precision average, which approximates to the average of the area under the precision-recall curve for a set of queries. In order to present a detailed result, this calculation is divided into several parts (tables). As the used corpus is bounded, documents that should be retrieved are well-known, and the recall measure can be calculated in each position (retrieved document) for each query. In the position where all relevant documents have been retrieved, the recall measure has a value of "1". The following table shows the recall value for each query.

Recall(R_{jk})

Q(q_j)	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}	d_{11}	d_{12}	d_{13}	d_{14}
q_1	0,17	0,33	0,33	0,5	0,67	0,83	1							
q_2	0,08	0,17	0,25	0,33	0,42	0,5	0,58	0,67	0,75	0,83	0,92	1		
q_3	0,14	0,29	0,43	0,57	0,71	0,86	0,86	0,86	1					
q_4	1													
q_5	1													
q_6	1													
q_7	0,33	0,33	0,33	0,33	0,33	0,67	1							
q_8	0,33	0,67	0,67	1										
q_9	0,5	0,5	1											
q_{10}	0,5	1												
q_{11}	0,33	0,67	1											
q_{12}	0,25	0,5	0,5	0,75	1									
q_{13}	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1				
q_{14}	0,07	0,14	0,21	0,29	0,36	0,43	0,5	0,57	0,64	0,71	0,79	0,86	0,93	1
q_{15}	0,11	0,22	0,33	0,44	0,56	0,67	0,78	0,89	1					
q_{16}	0,13	0,25	0,25	0,25	0,38	0,38	0,5	0,63	0,63	0,75	0,88	0,88	0,88	1
q_{17}	0,33	0,67	1											
q_{18}	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1				
q_{19}	0,2	0,4	0,4	0,4	0,4	0,4	0,6	0,8	1					
q_{20}	0,25	0,5	0,75	1										
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

q_j: query number j; d_k: document retrieved in position k

The following table shows precision values.

Precision(R_{jk})

Q(q_j)	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}	d_{11}	d_{12}	d_{13}	d_{14}	$\sum P(R_{jk})$	$\frac{1}{m_j}\sum P(R_{jk})$
q_1	1	1	0,67	0,75	0,8	0,83	0,86	0	0	0	0	0	0	0	5,91	0,84
q_2	1	1	1	1	1	1	1	1	1	1	1	1	0	0	12	1
q_3	1	1	1	1	1	1	0,86	0,75	0,78	0	0	0	0	0	8,38	0,93
q_4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
q_5	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
q_6	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
q_7	1	0,5	0,33	0,25	0,2	0,33	0,43	0	0	0	0	0	0	0	3,05	0,44
q_8	1	1	0,67	0,75	0	0	0	0	0	0	0	0	0	0	3,42	0,85
q_9	1	0,5	0,67	0	0	0	0	0	0	0	0	0	0	0	2,17	0,72
q_{10}	1	1	0	0	0	0	0	0	0	0	0	0	0	0	2	1
q_{11}	1	1	1	0	0	0	0	0	0	0	0	0	0	0	3	1
q_{12}	1	1	0,67	0,75	0,8	0	0	0	0	0	0	0	0	0	4,22	0,84
q_{13}	1	1	1	1	1	1	1	1	1	1	0	0	0	0	10	1
q_{14}	1	1	1	1	1	1	1	1	1	1	1	1	1	1	14	1
q_{15}	1	1	1	1	1	1	1	1	1	0	0	0	0	0	9	1
q_{16}	1	1	0,67	0,5	0,6	0,5	0,57	0,63	0,56	0,6	0,64	0,58	0,54	0,57	8,95	0,64
q_{17}	1	1	1	0	0	0	0	0	0	0	0	0	0	0	3	1
q_{18}	1	1	1	1	1	1	1	1	1	1	0	0	0	0	10	1
q_{19}	1	1	0,67	0,5	0,4	0,33	0,43	0,5	0,56	0	0	0	0	0	5,38	0,6
q_{20}	1	1	1	1	0	0	0	0	0	0	0	0	0	0	4	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
$MAP(Q) = \frac{1}{\lvert Q\rvert}\sum_{j=1}^{\lvert Q\rvert}\frac{1}{m_j}\sum_{k=1}^{m_j} Precision(R_{jk})$																**0.895**

q_j: query number j; d_k: document retrieved in position k

With a MAP value of 0.895, the proposed strategy ensures that relevant documents are in the top of retrieved documents in most cases. Calculating in ascending order the average of precision value in each point at which each relevant document is retrieved, and then calculating the average of this average of precision values for a set of queries (i.e. the MAP value), a high value only is obtained if the best results are in the top positions of the answer provided by the system, aim sought in any information retrieval system.

7 Conclusions and future work

In this paper, we have presented an integration of information retrieval technologies, domain ontologies, and organizational knowledge management systems. The exposed strategies are straightforward domain-independent approaches for annotating structured and unstructured information sources. These strategies use a domain ontology to add domain semantics. They do not require additional linguistic analysis methods, such as collocation patterns and linguistic patterns, to capture domain-relevant concepts. The results of the tests have been highly satisfactory both for quantitative results (MAP value) and for the complexity of the queries that the systems could answer.

Moreover, proposed strategies are strongly dependent of the domain ontology and its completeness; therefore, the architecture also includes an ontology evolution strategy (which is detailed in [30]). Now, this ontology evolution strategy works with simple concepts only composed of nouns. In future work, we will extend this strategy to allow the handling of complex concepts. In this new strategy, we will experiment with statistical methods such as C-value/NC-value methods to determine the relevance of terms with respect to the domain corpus.

References

1. Ale, M.A.: An Organizational Knowledge Management Conceptual Model. PhD in information systems, National Technological University (2009), iSBN 978-987-05-8131-4
2. Ale, M.A., Chiotti, O., Galli, M.R.: Enterprise Knowlegde Management for Emergent Organizations: An Ontology-Driven Approach. IGI Global, Hershey (2008)
3. Ale, M.A., Gerarduzzi, C., Chiotti, O., Galli, M.R.: Organizational knowledge sources integration through an Ontology-Based approach: The Onto-DOM architecture. In: Lytras, M.D., Carroll, J.M., Damiani, E., Tennyson, R.D. (eds.) World Summit On The Knowledge Society. Lecture Notes in Computer Science, vol. 5288, pp. 441–450. Springer (2008)
4. Aronson, A.: Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. In: Proceedings of the AMIA Symposium. p. 17. American Medical Informatics Association (2001)
5. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. Addison Wesley, 1st edn. (May 1999)
6. Bolloju, N., Khalifa, M., Turban, E.: Integrating knowledge management into enterprise environments for the next generation decision support. Decision Support Systems 33(2), 163–176 (2002)

7. Bonifacio, M., Bouquet, P., Cuel, R.: The role of classification(s) in distributed knowledge management. In: Proceedings of 6th International Conference on Knowledge-Based Intelligent Information Engineering Systems & Allied Technologies Special Session on Classification. pp. 448–452 (2002)
8. Broekstra, J., Kampman, A., van Harmelen, F.: Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. In: Horrocks, I., Hendler, J. (eds.) Proceedings of the first International Semantic Web Conference (ISWC 2002). pp. 54–68. No. 2342 in Lecture Notes in Computer Science, Springer Verlag, Heidelberg Germany, Sardinia, Italy (June 9 – 12, 2002)
9. Cimiano, P., Handschuh, S., Staab, S.: Towards the self-annotating web. In: Proceedings of the 13th international conference on World Wide Web. pp. 462–471. WWW '04, ACM, New York, NY, USA (2004)
10. Cunningham, H.: Gate, a general architecture for text engineering. Computers and the Humanities 36, 223–254 (2002)
11. Dill, S., Eiron, N., Gibson, D., Gruhl, D., Guha, R., Jhingran, A., Kanungo, T., Rajagopalan, S., Tomkins, A., Tomlin, J., et al.: SemTag and Seeker: Bootstrapping the semantic web via automated semantic annotation. In: Proceedings of the 12th international conference on World Wide Web. pp. 178–186. ACM (2003)
12. Dingli, A., Ciravegna, F., Wilks, Y.: Automatic semantic annotation using unsupervised information extraction and integration. In: Proceedings of SemAnnot 2003 Workshop (2003)
13. Ermine, J.: Challenges and approaches for knowledge management in companies. In: Workshop in Knowledge Management: Theory and Applications. pp. 5–11. Lyon, France (2000)
14. Feldman, R., Sanger, J.: The text mining handbook. Cambridge University Press (2007)
15. Fensel, D., Angele, J., Decker, S., Erdmann, M., Schnurr, H., Staab, S., Studer, R., Witt, A.: On2broker: Semantic-based access to information sources at the WWW. In: Proceedings of the World Conference on the WWW and Internet (WebNet 99). pp. 366–371 (1999)
16. Fensel, D.A.: Ontologies:: A Silver Bullet for Knowledge Management and Electronic Commerce. Springer, 2nd edn. (Dec 2003)
17. Gschwandtner, T., Kaiser, K., Martini, P., Miksch, S.: Easing semantically enriched information retrieval–an interactive semi-automatic annotation system for medical documents. International Journal of Human-Computer Studies 68(6), 370 – 385 (2010)
18. Hackbarth, G., Grover, V.: The knowledge repository: Organizational memory information systems. Information Systems Management 16(3), 21–30 (1999)
19. Handschuh, S., Staab, S., Ciravegna, F.: S-cream semi-automatic creation of metadata. In: Gmez-Prez, A., Benjamins, V. (eds.) Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web, Lecture Notes in Computer Science, vol. 2473, pp. 165–184. Springer Berlin / Heidelberg (2002)
20. Kiryakov, A., Popov, B., Terziev, I., Manov, D., Ognyanoff, D.: Semantic annotation, indexing, and retrieval. Web Semantics: Science, Services and Agents on the World Wide Web 2(1), 49 – 79 (2004)
21. Kiyavitskaya, N., Zeni, N., Cordy, J.R., Mich, L., Mylopoulos, J.: Cerno: Lightweight tool support for semantic annotation of textual documents. Data & Knowledge Engineering 68(12), 1470 – 1492 (2009)
22. Luke, S., Spector, L., Rager, D., Hendler, J.: Ontology-based Web agents. In: Proceedings of the first international conference on Autonomous agents. pp. 59–66. ACM (1997)

23. Malone, D.: Knowledge management: A model for organizational learning. International Journal of Accounting Information Systems 3(2), 111 – 123 (2002)
24. Manning, C., Schütze, H.: Foundations of statistical natural language processing, vol. 59. MIT Press (1999)
25. McAdam, R., McCreedy, S.: The process of knowledge management within organizations: a critical assessment of both theory and practice. Knowledge and Process Management 6(2), 101–113 (1999)
26. Reeve, L., Han, H.: Survey of semantic annotation platforms. In: Proceedings of the 2005 ACM symposium on Applied computing. pp. 1634–1638. SAC '05, ACM, New York, NY, USA (2005)
27. Schreiber, G., Akkermans, H., Anjewierden, A., de Hoog, R., Shadbolt, N., de Velde, W.V., Wielinga, B.: Knowledge Engineering and Management - The CommonKADS Methodology. The MIT Press, Cambridge, Massachusetts; London, England (1999)
28. Souza, R.G.S.: Agent-oriented constructivist knowledge management. Ph.D. thesis, University of Twente, Enschede (2006)
29. Toledo, C.M., Ale, M., Chiotti, O., Galli, M.R.: An agent-based architecture for ontology-driven knowledge management. In: The V International Conference on Knowledge, Information and Creativity Support Systems. pp. 47–54. Thailand (2010)
30. Toledo, C.M., Ale, M., Chiotti, O., Galli, M.R.: An agent-based ontology evolution strategy for ontology-driven knowledge management systems. In: XXXVI Latin American Informatics Conferences (CLEI). Paraguay (2010)
31. Uren, V., Cimiano, P., Iria, J., Handschuh, S., Vargas-Vera, M., Motta, E., Ciravegna, F.: Semantic annotation for knowledge management: Requirements and a survey of the state of the art. Web Semantics: Science, Services and Agents on the World Wide Web 4(1), 14 – 28 (2006)
32. Vargas-Vera, M., Motta, E., Domingue, J., Lanzoni, M., Stutt, A., Ciravegna, F.: MnM ontology driven semi-automatic or automatic support for semantic markup. In: Gómez-Pérez, A., Benjamins, V.R. (eds.) Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web, Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management, EKAW02. LNAI, vol. 2473, pp. 379–391. Springer Verlag, Berlin (2002)