Revista EIA, ISSN 1794-1237 / Year XI / Volume 11 / Issue N.22 / July-December 2014 / pp. 45-58
Technical-scientific biannual publication / Escuela de Ingeniería de Antioquia —EIA—, Envigado (Colombia)

PROPOSAL FOR SEMANTICALLY EXTENDING THE INFORMATION RETRIEVAL PROCESS

Oswaldo Solarte Pabón[1]
Martha Millán[2]

ABSTRACT

In classic information retrieval models, documents are represented by a set of terms or keywords. A disadvantage of this representation is that the query results are limited only to the frequency of occurrence of terms. It does not consider the meaning of terms or semantic relationships that may exist between the documents. One alternative to solve this problem is using Semantic Web technologies in order to assign data a well-defined meaning. This article describes a proposal to extend the information retrieval process using Semantic Web technologies. Documents are semantically enriched with annotations that are obtained from a domain ontology. The extended semantic retrieval takes into account both keywords expressed in the user query as well as their meaning, which is represented through ontology. Extended semantic retrieval improves results in terms of precision and recall when compared with results obtained with classic information retrieval.

KEYWORDS: Information retrieval (IR), Semantic Web, ontologies, semantic annotations

PROPUESTA PARA EXTENDER SEMÁNTICAMENTE EL PROCESO DE RECUPERACIÓN DE INFORMACIÓN

RESUMEN

En los modelos clásicos de recuperación de información los documentos se representan mediante un conjunto de términos o palabras clave. Una desventaja de esta representación es que los resultados de una consulta se limitan solo a la frecuencia de aparición de los términos. No se tiene en cuenta el significado de los términos ni las relaciones semánticas que puedan existir entre los documentos. Una de las alternativas para resolver este problema es usar tecnologías de la Web semántica con el objetivo de asignarle a los datos un significado bien definido. En este artículo se describe una propuesta para extender el proceso de recuperación de información usando tecnologías de la Web semántica. Los documentos se enriquecen semánticamente por medio de anotaciones que se obtienen a partir de una ontología de dominio. La recuperación de información extendida semánticamente tiene en cuenta tanto las palabras clave expresadas en la consulta del usuario como también su significado, el cual se representa mediante una ontología. La recuperación

[1] Systems Engineer, Master in Eng. Systems and Computer Science, Universidad del Valle. Professor, School of Computer and Systems Engineering, Universidad del Valle.

[2] Degree in Mathematics and Physics, Universidad del Valle. PhD in Computer Science, Universidad Politécnica de Madrid - Spain. Professor of the School of Computer and Systems Engineering, Universidad del Valle.

Correspondence author: Solarte-Pabón, O. (*Oswaldo*). Ed. 331 2° piso Ciudad Universitaria Meléndez, Universidad del Valle. Tel: (572) 321 22 83. Email: oswaldo.solarte@correounivalle.edu.co

Paper history:
Paper received on:22-IV-2013 / Approved: 19-VIII-2014
Available online: December 30 2014
Open discussion until December 2015

DOI: http://dx.doi.org/10.14508/reia.2014.11.22.49-63 reia.2014.11.22.51-65

de información extendida semánticamente mejora los resultados en términos de *precision* y *recall* en comparación con los obtenidos en la recuperación de información clásica.

PALABRAS CLAVE: recuperación de información (IR); Web semántica; ontologías; anotaciones semánticas.

PROPOSTA PARA ESTENDER SEMÂNTICAMENTE O PROCESSO DE RECUPERAÇÃO DE INFORMAÇÃO

RESUMO

Em modelos convencionais de recuperação de informação os documentos são representados por um conjunto de termos ou palavras-chave. A desvantagem desta representação é que os resultados da consulta são limitados apenas para a frequência de ocorrência dos termos. Não é levado em conta o significado dos termos nem as relações semânticas que possam existir entre os documentos. Uma das alternativas para resolver este problema é a utilização de tecnologias da Web Semântica, com o objetivo de dar aos dados um significado bem definido. Este artigo descreve uma proposta para estender o processo de recuperação de informação utilizando tecnologias de web semântica. Os documentos são semanticamente enriquecidos por anotações que são derivados de uma ontologia de domínio. A recuperação de informação estendida semanticamente leva em conta tanto as palavras-chave expressas na consulta do usuário, bem como o seu significado, que é representado por uma ontologia. A recuperação da informação estendida semântica melhora os resultados em termos de precisão e recall comparados com os obtidos na recuperação de informação convencional.

PALAVRAS-CHAVE: recuperação da informação (IR); Web Semântica; ontologias; anotações semânticas.

1. INTRODUCTION

Classic information retrieval models are widely used to support the development of search tools. In these models, documents are represented as a set of terms (Baeza, 1999). Systems based on these models allow the user to search for information through a query mechanism based on keywords. For a given query, a set of documents is returned that in some way satisfies the user's information needs (TrivikRam, 2007). When a query is made, these models work with the frequency of the terms' appearance to assign importance to the documents. The results obtained show an order of relevance with regards to the query terms. Despite the fact that many systems have been built using classic information retrieval models, these systems only use the frequency of the terms' appearance without taking their meaning into account (Wei et al., 2007).

One proposal for improving the effectiveness of the results obtained by information retrieval systems is to include the use of Semantic Web technologies. The Semantic Web is an extension of the current World Wide Web in which a well-defined meaning is given to data, making it easier for computers and people to work cooperatively (Lee et al., 2001). One of the advantages offered by the Semantic Web is that it considers the meaning of words within documents. By offering the possibility of searching for information while bearing in mind the data's semantic aspects, better results can be obtained for a query.

This article presents a proposal for extending the information retrieval process using Semantic Web technologies. Documents are semantically enriched through annotations obtained from a domain ontology. The user's queries are expanded through the properties of annotation and instances defined in the ontology. In the document search process, classic information retrieval is combined with retrieval based on semantic annotations. Extended information retrieval with semantic annotations improves results in terms of precision and recall when compared to results obtained with classic information retrieval.

The remainder of the article is organized as follows: section 2 presents some related studies in which Semantic Web technologies are used to improve the

information retrieval process; section 3 describes a model for semantically extending the information retrieval process through the use of semantic annotations; section 4 presents an automatic mechanism for semantically annotating text documents; section 5 describes a strategy for searching for semantically enriched documents; section 6 presents the implementation of a prototype and the tests that have been carried out; finally, section 7 contains conclusions and future study.

2. RELATED STUDIES

In recent years, different studies have aimed to improve information retrieval using Semantic Web technologies (Vallet et al., 2005), (Castells et al., 2007), (Bhagdev et al, 2008). These studies have been framed within the field of semantic search, making reference to systems that use Semantic Web technologies to improve the different parts of the information retrieval process. According to Mangold (2007), semantic search is defined as a document retrieval process that takes advantage of knowledge of a domain and can be formalized through an ontology. According to Wei et al. (2008), the goal of semantic search is to improve the conventional techniques and methods of information retrieval. Nagypal (2007) classifies search systems into two categories: those that focus on instance retrieval based on an ontology and those that focus on document retrieval. This article focuses on the second category. Its purpose is to improve information retrieval for text documents using ontologies and semantic annotations. Semantic search systems aimed at improving document retrieval can be seen as an extension of classic information retrieval. In these systems, documents are semantically annotated based on a domain ontology. The retrieval process is carried out by making the users' queries coincide with the semantic annotations pulled from the documents (Wei et al., 2008).

2.1. Criteria for analyzing semantic search systems

Various authors, including Manglod (2007), Wei et al., (2008), and Strasunskas, D. & Tomassen S. (2010), have classified different semantic search systems based on a set of criteria that allow them to analyze the systems' most important characteristics. In order to facilitate the analysis of semantic search systems for document retrieval, we have selected the following criteria: the system's level of transparency, the language used to represent knowledge (RDF, OWL, DAML+OIL), the semantic annotation mechanism, and how the user makes queries. The level of transparency refers to how the user interacts with the semantic search system. This interaction can be invisible if the semantic capabilities are hidden from the user, interactive if the system asks the user for feedback to make changes to the query, or hybrid if it is a combination of the two. The annotation mechanism can be manual, semiautomatic, or automatic. Manual annotation is a difficult and costly process in terms of time and people required to carry it out (Corcho, 2006). Semiautomatic annotation requires minimum user intervention, and the annotations are made based on suggestions from an automatic process (Oren et al., 2006). In automatic annotation, the semantic annotations are made almost without user intervention and reduce costs in terms of time and users required to make the annotations.

The way in which users make queries to the semantic search system can be: based on keywords, in forms, in natural languages, or based on a formal query language (e.g. Sparql [1]). Keyword-based systems are characterized by their ease of use. Systems based on forms graphically present the user with parts of the ontology structure so that he or she can select the classes with which to perform the search. The disadvantage of these systems is that they are not very flexible given that the user can only select the elements shown on the form (Uren et al., 2007). Also, the user spends a lot of time navigating the ontology structure. For their part, natural language-based systems offer precise answers to the users' queries (query answering). Finally, some semantic search systems require that the query be expressed using a formal query language. This can be a disadvantage because it represents a high level of complexity for users.

2.2. Proposals that use Semantic Web technologies to improve information retrieval

One of the first proposals aiming to improve the process of information retrieval using Semantic Web

1 http://www.w3.org/TR/rdf-sparql-query

technologies was presented by Shah et al. (2002). In this proposal, documents are enriched with semantic annotations that are automatically obtained by applying information extraction techniques. The annotations are stored within the document itself, and a query can be expressed through keywords or using the query language DQL (DAML+OIL query language). Popov et al. (2004) and Kiryakov et al. (2005) describe KIM, a framework that also uses an automatic semantic annotation for documents. The annotations are represented as links between concepts included in the document and classes in an ontology. In contrast to the proposal above, these semantic annotations are stored in a knowledge base that is separate from the documents and is represented through RDF triples[2]. When a query is launched in KIM, first the instances in the ontology that are associated with the query terms are searched for, then the documents annotated with these instances are retrieved. In KIM, the relevance of the annotations is not considered, and it is therefore difficult to apply a ranking algorithm that allows documents to be ordered according to the relevance of the semantic annotations.

For their part, Vallet et al. (2005) and Castells et al. (2007) adapt the vector space model to facilitate semantic search for documents based on the relevance of the annotations. The authors propose a ranking algorithm to calculate the semantic annotations' degree of relevance. The degree of relevance depends on the frequency of ontology classes with which the documents are annotated. This ranking algorithm orders the documents based on the relevance of the semantic annotations. The system takes a query expressed in Sparql as input and returns a list of ontology instances. Based on these instances, the query is expanded, exploring the ontology class hierarchies. Finally, the documents annotated with the ontology instances are retrieved and ordered according to the similarity between the query and the semantic annotations. According to the authors, semantic search improves the results in comparison to a search based solely on classic models of information retrieval. However, semantic search can fail when the semantic annotations are incomplete and do not cover all the information in a document. One disadvantage of this proposal is that the query must be expressed in Sparql, which can represent a high level of complexity for the system's users.

Bhagdev et al. (2008) and Bikakis (2010) apply the concept of hybrid search that combines results of a search based on classic IR models with results of a search based on semantic annotations. The search based on classic IR models (e.g. vector space model) only considers the frequency with which the keywords appear. However, the semantic annotations search can fail when the ontology used as a basis for the annotation does not cover all the semantics of a document. Hybrid search deals with these problems by combining a search based on classic IR models with one based on semantic annotations. This is an advantage given that the results are improved in terms of precision and recall in average cases. However, one disadvantage of this proposal is how the user interacts with the system; he or she must navigate the hierarchical structure of the ontology and manually select the classes which will orient the search process. The user must invest a large amount of time in selecting the classes and must also know the structure of the ontology.

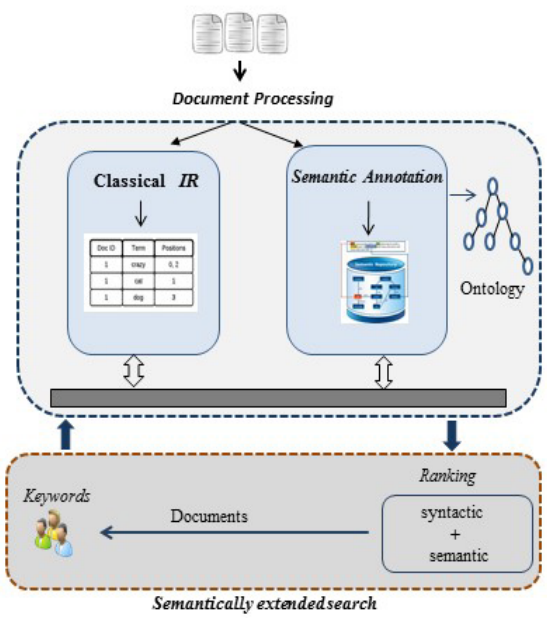
Finally, Rodriguez-García et al. (2014a, 2014b) describe a platform for semantically enriching the discovery of cloud services. This platform uses the descriptions of the cloud services as documents, and based on these, automatically creates semantic annotations. In the annotation process, multiple ontologies and document formats can be used, and the semantic annotations are indexed by adapting the classic vector space model. For each document, a vector is created in which each dimension represents the level of relevance of a concept in the ontology for said document. The platform allows for semantic search of documents using keywords, which makes user interaction easier. One of the most interesting aspects of this proposal is that it proposes a module for supporting the evolution of ontologies. This aims to enrich, improve, and increase the knowledge represented in the ontologies. To support this process, an algorithm is proposed to search for information in Wikipedia for the terms that are not represented in the ontology. These terms are used to search Wikipedia for articles that coincide with the search terms, and then a new concept that contains synonyms both in English and Spanish is created and added to the ontology.

2 http://www.w3.org/RDF/

3. MODEL FOR SEMANTICALLY EXTENDING THE INFORMATION RETRIEVAL PROCESS

According to Baeza (1999), the information retrieval process includes stages like information modeling, indexation, query, and results ranking. In the model in **Figure 1**, this process is extended with a semantic annotation stage in order to give meaning to the terms in a document. According to Bontcheva et al. (2006), semantic annotations allow links to be created between the entities present in a text and their descriptions defined in a semantic structure as an ontology. This model differs from other semantic search proposals in how the user interacts with the system. In this model, the user expresses queries using only keywords without having to manually select ontology classes or know the ontology's structure. The user also does not need to know formal query languages in order to access the semantic annotations. According to Tran et al. (2009), the user is used to expressing queries through usable query interfaces that are generally based on keywords. The model in **Figure 1** is made up of two main components: document processing, which is shown in the upper section of the figure, and semantically extended search in the lower section.

Figure 1. Model for semantically extending information retrieval.



The document process is divided into two modules. In the first (classic IR), documents are processed through the application of techniques like dividing the text into tokens, eliminating stopwords, and applying stemming algorithms. In this module, each document is represented by a set of terms using the classic vector space model (Salton et al., 1975). A detailed description of this module is presented in section 4. The second module (semantic annotation) undertakes the process of making semantic annotations based on a domain ontology. In the semantic annotation module, the documents are represented by a set of annotations which have a well-defined meaning in the ontology. Section 5 describes the semantic annotation process in detail.

Semantically extended search makes it possible to search for documents combining classic information retrieval with a semantic annotation search. The results obtained separately are mixed and shown to the user with a hybrid ranking shown in Formula 1. The hybrid ranking is calculated by combining the search relevance based on classic IR techniques (ir-score) with the search relevance based on semantic annotations (semantic-score).

$$hybrid\text{-}score = \lambda(ir\text{-}score) + \omega(semantic\text{-}score) \quad (1)$$

Factors λ and ω represent the degree of importance of the search based on classic *IR* techniques and on semantic annotations, respectively. The values of λ and ω are between 0.0 and 1.0. If the value for λ and ω is 0.5, this means that both types of search have the same importance. The use of these factors is explained in detail in section 5, which covers semantically extended document search.

3.1. Document processing based on classic IR techniques

In this proposal, classic IR techniques refer to the use of the vector space model for representing documents as vectors of terms (Salton et al., 1975) and also to the fact that the relevance of a term only depends on the frequency with which it appears in the documents. This processing is divided into two phases: analysis and indexation. The first applies a series of techniques such as the division of the text into tokens, the eliminations of accent marks and stopwords, and the reduction of terms

to their roots using the stemming algorithm proposed by Porter (1997). The indexation phase takes the set of terms obtained in the previous phase as input and represents them using the vector space model. In this model, a document is represented as a vector of terms. Each term is associated with a degree of relevance that is calculated using the TF-IDF algorithm (Manning et al., 2008). Relevance depends only on term frequency (TF) in the document and the inverse document frequency (IDF), that is, the occurrence of the term in the collection of documents. The result of this phase is an index in which each document has a corresponding vector, and each vector component represents the degree of relevance a term has for the document.

4. SEMANTIC ANNOTATION OF DOCUMENTS

The semantic annotation process was divided into two parts. In the first part, we implemented the ontology that is used as a basis for annotating the text documents. In the second part, we developed an automatic semantic annotation mechanism based on the API of the tool GATE[3]. A description of each of these parts follows.

4.1. Base ontology for the semantic annotation process

During the semantic annotation process, we used an ontology proposed by ACM[4] which describes the domain of computer science. It was implemented in Protegé using the language *OWL*. Each ontology class represents an area of knowledge in this domain. Several annotation properties were defined for each class: *english_name*, *spanish_name*, and *related_content*. The first two were used to associate a tag to each class both in English and Spanish, respectively. The *related_content* property is used to describe synonyms or alternative ways to textually represent concepts in the domain. **Table 1** shows an example of the annotation properties and their values defined for the "*#Association_rules*" class. All the values in this table are semantically related.

3 http://gate.ac.uk/

4 http://www.computer.org/portal/web/publications/acmtaxonomy

During the semantic annotation process, all the previously defined annotation properties are considered, as well as class instances. That is, the different ways of representing a domain concept in the text to create links between documents and ontology classes are considered.

Table 1. Some annotation properties defined in the ontology

Property	Value
# english_name	Association rules
# spanish_name	Reglas de asociación
# related_content	A priori algorithm
# related_content	FP-growth algorithm

4.2. Automatic document annotation based on GATE

GATE is a text processing framework, and its functionalities can be accessed through a graphic interface or by using an API that allows it to be integrated into other applications. In this proposal, the API is embedded in order to create an automatic document annotation mechanism. To complete semantic annotation process, language and processing resources were created. The language resources allow us to define the ontology that will guide the annotation process and the documents to be annotated. The processing resources analyze the documents and create the semantic annotations. **Figure 2** shows the processing resources that were defined in GATE to automatically complete the annotation process. The sentence splitter resource divides the texts into declarations, which can be complete sentences or phrases, and the tokenizer resource divides the text into tokens. The POS tagger and morphological analyzer resources are used to assign each word in the text a grammatical category (e.g. verb, article, adverb, pronoun). Finally, the OntoRoot Gazetteer resource allows us to associate the concepts found in the text with the ontology classes. The result of this process is a list of semantic annotations that represent links between the text and ontology classes.

Figure 3 shows the algorithm used for automatic semantic annotation. This algorithm receives as input a list of documents that will be annotated (CorpusDocuments) and a file (gateApp) that describes the processing resources previously defined in GATE. For

each document, its identifier (docID) is obtained, and then the annotation process defined in the gateApp file is carried out (line 5). The gateApp.execute method returns the set of completed semantic annotations to the document. This set is then analyzed (lines 7-12), and for each annotation, ontology attributes are received, such as: class (concept), URI, and the annotation property with which the document was annotated. The algorithm returns a list containing the annotations for each of the documents processed.

In the semantic annotation process presented in **Figure 3**, a document can be annotated with several ontology classes, and one class can be used to annotate different documents. Given this situation, some semantic annotations can be more relevant than others. Therefore, after the algorithm in **Figure 3** completes its execution, it proceeds to calculate the relevance of the semantic annotations. The relevance allows it to identify which ontology classes are most important for each document. The degree of semantic relevance is obtained using Formula 2 and is based on the proposal of Castells et al. (2007).

$$W_{i,j} = \frac{freq_{i,j}}{max_l \ freq_{l,j}} * \log \frac{N}{n_i} \qquad (2)$$

The weight $W_{i,j}$ is the degree of relevance that annotation ***i*** has for document ***j***. In order to calculate this weight we first calculate the frequency of the class ($f_{i,j}$), which is the number of semantic annotations document ***j*** has with regards to class ***i*** in the ontology. This frequency is normalized by dividing it among the maximum frequency ($maxl\,freq_{l,j}$). The frequency of the class ($f_{i,j}$), is multiplied by the inverse frequency of the document, in which *N* is the number of documents in the collection and n_i is the number of documents annotated with class ***i***. A semantic annotation is represented with three attributes: the document identifier (*doc_id),* the class of the ontology with which the annotation was made (*ontology_class*) and the degree of relevance between the class and the document (relevance). After the relevance is calculated, each document has only one annotation for each class of the ontology and its respective degree of relevance. Finally, the semantic annotations are stored in the form of RDF triples using the Jena framework.

Table 2 shows an example in which a semantic annotation (*annotation_1*), has been created and linked with a document (*document_3*), through the "*linked_to*". property. Thereby, we can say that *document_3* was semantically annotated with the *"Database_Systems"* class, and this annotation has a degree of relevance of 0.25.

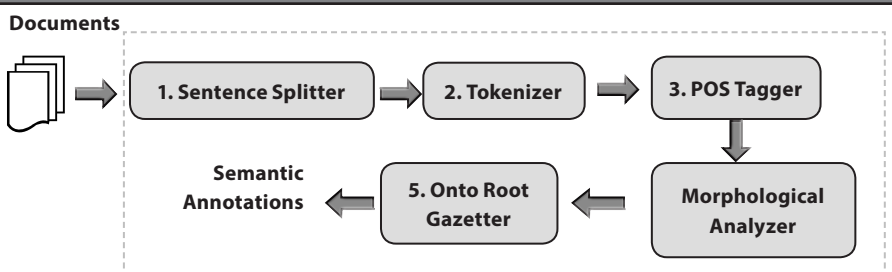
Table 2. Semantic annotations expressed in triplets

Subject	Property	Object
annotation_1	*linked_to*	*document_3*
annotation_1	*ontology_class*	*Database_Systems*
annotation_1	*weight*	*0.25*

4.3. Extended semantic annotation

In this stage, a hierarchical structure of the ontology and the concept of semantic distance are used to describe new classes with which to relate a document. According to Nesic et al. (2010), semantic distance can be understood as the number of jumps that must be made in the ontology to get from one node to another. In this case, it is assumed that each node represents an ontology

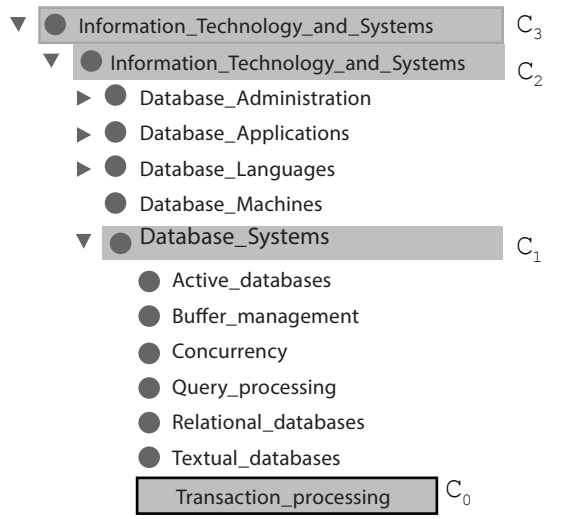
Figure 2. Processing resources used in GATE to complete semantic annotation



class. To discover new classes, we start from the classes that were obtained with the GATE tool, which are called base classes. From these base classes, we explore the hierarchical structure of the ontology, and using the rdf:subClassOf property, we obtain the preceding classes for the base class. Semantic annotations are created for the newly discovered classes, and they receive a degree of relevance to the base class. The smaller the distance between the base class and the newly discovered class, the greater the relevance of the new annotation. On the other hand, the greater the semantic distance, the lower the degree of relevance for the new annotation.

Furthermore, the hierarchy of concepts in the ontology can be very large, which implies that there will be very long distances between one concept and another. To avoid making annotations between concepts that are very distant in the hierarchical structure, we must define a limit for semantic distance. In this proposal, the limit can be specified in two ways: the first is to define a value manually each time an annotation process is begun; the second is to configure a default limit value. Samper J. et al. (2008) recommend using a limit less than or equal to three, given that concepts whose distance is greater than three do not have a well-defined semantic relationship due to how the hierarchy of concepts is built. **Figure 4** shows a part of the ontology that was used during the semantic annotation process.

Figure 4. Hierarchical structure of the ontology



If the *"Transaction_procesing"*, class, identified by tag C_0,is taken as a base, the semantic distance between C_0 *and* C_1 is less than the semantic distance between C_0 and C_3. Based on this semantic distance, we can say that for the *"Transaction_procesing"* class, the *"Database_Systems"* class is more relevant than the *"Information_Tecnology_and_Systems"class.* Extended semantic annotation creates new annotations with their respective degree of relevance, which is calculated using **Formula 3**. The degree of relevance of a related class (*Wrc*), depends on the semantic distance (*Sem-Distance*) and the degree of relevance of the base class

Figure 3. Algorithm for making semantic annotations using GATE API.

```
1  PROCEDURE  semanticAnnotator (CorpusDocuments, gateApp)
2
3       FOR EACH document  IN CorpusDocuments
4           docId  =  document.getID ();
5           gateAnnotations = gateApp.execute(document);
6
7           FOR EACH  annotation  IN gateAnnotations
8              concept = annotation.getConcept ();
9              uri =  annotation.get Uri ();
10             property =  annotation.getProperty ();
11             annList.add  (docID, concept, uri, property);
12          END
13      END
14     return annList;
15 END
```

(*Wbc*), which is previously obtained using **Formula 2** (Section 4.2). The degree of relevance *Wrc* will be higher the smaller the semantic distance between the classes.

$$Wcr = Wbc * \beta^{-SemDistance}; \quad (3)$$

5. SEMANTICALLY EXTENDED USER QUERIES

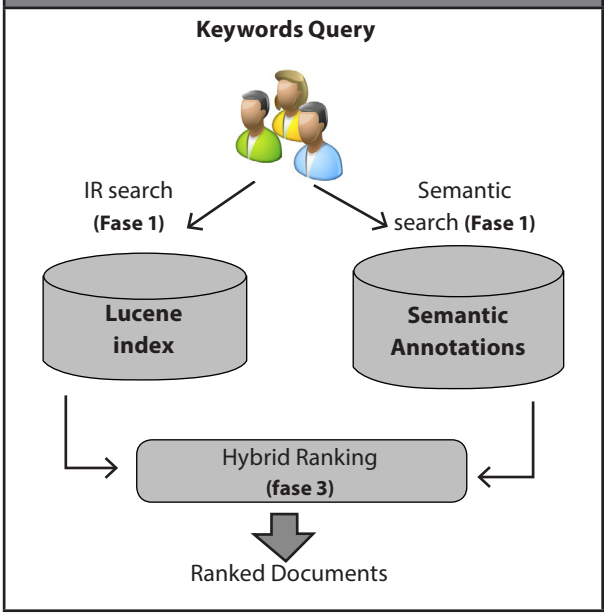
A query is processed in three phases, as shown in **Figure 5**. In phase 1, the documents are retrieved based on the vector space model and TD-IDF metric (classic IR techniques). In phase 2, retrieval is done based on semantic annotations. The two previous phases can be executed simultaneously since they are processed in separate repositories. In phase 3, the results obtained in the first two phases are mixed using a hybrid ranking algorithm and are shown to the user.

Phase 1: This is called the traditional search phase since it is based on classic IR techniques. Documents are represented as vectors and are retrieved considering only term frequency. The result is an ordered list of documents whose relevance is calculated by comparing the similarity between the query vector and the vectors of each document. To implement this phase, we used the Apache Lucene[5] API, an open source tool that implements the vector space model.

Phase 2: In this phase, the document search is based on semantic annotation and is completed in three steps: transformation of keywords into a set of classes in the ontology, a search of the annotated documents with these classes, and ordering of the documents retrieved, according to the degree of relevance of the semantic annotations.

Keywords are transformed into ontology classes through a concept index that stores pairs in the form (*class, text*). The class element corresponds to the URI of the ontology class, and the text element contains the text that has been extracted from the annotation properties of the class. The concept index is created beforehand with a pre-processing of the ontology and allows for automatic receipt of the classes that will orient the search process. The users express queries using key words, while the semantic annotations are stored in the form of RDF triples. Considering that forms of expressing the user queries and the semantic annotations are different, it was necessary to create the concept index which allows interpretation of a query expressed in keywords to be related with the ontology classes. For example, if the query expressed by the user is "*Rules of association in data mining*," the ontology classes obtained are those shown in **Table 3**. Once the classes are obtained, a vector that represents the query is created to be used later to calculate the similarity between the documents and the query. If the user has expressed the query in Spanish, the same ontology classes would be obtained (shown in **Table 2**). This is owed to the *english_name* and *spanish_name* annotation properties of each class.

Figure 5. Semantically extended user queries



After transforming the keywords into ontology classes, a Sparql query is automatically generated. Each class obtained in the previous step is added to the WHERE clause of the query. **Figure 6** shows part of the query generated. The Sparql query returns a list of semantic annotations in which each element on the list contains the document identifier, the ontology class, and the semantic annotation's degree of relevance. For example, the query "*Rules of association in data mining*" is semantically expanded with keywords like "*A priori algorithm*" and "*FP-growth algorithm*" because these concepts are semantically relates to the "*#Association_rules*" class, as shown in **Table 3**. After the list of annotations that coincide with the search expressed by

5 http://lucene.apache.org/core/

the user is returned, the relevance between the query and the documents is calculated. Both the query and the documents are represented as vectors with each vector position corresponding to an ontology class. The relevance is calculated using cosine similarity (Tan P.N et al., 2006). Semantic annotation retrieval also expands the query with related ontology classes. Beginning from the query vector classes, semantically related ontology classes are searched for. This expansion offers more search possibilities to the user, such as related document search or document recommendation.

Table 3. Transformation of a query into classes

Ontology class	Degree of relevance
#Association_rules	1.00
#Data_mining	0.90
#Mining_methods_and_ algorithms	0.75
#Text mining	0.57
#Web_mining	0.52

***Fase 3*:** Phase 3: In this phase, the documents obtained in the classic IR technique search are combined with the documents obtained in the semantic annotation search (phases 1 and 2). The documents are combined by applying a hybrid ranking mechanism which considers a factor of importance for each type of search, as shown in Formula 1. Factors λ and ω are adjusted depending on the query conditions:

Condition 1: The query can be completely represented with the ontology concepts (classes, annotation properties, instances). More importance is placed on semantic annotation search ($\omega > \lambda$). In this case $\omega = 0.7$ and $\lambda = 0.3$

Condition 2: The query cannot by completely represented with the ontology concepts. The semantic importance factor will have a lower value, so that ($\omega < \lambda$). In this case $\omega = 0.4$ and $\lambda = 0.6$

The values of λ and ω were obtained based on supervised tests built on a set of documents and based on user queries completely and partially represented using the ontology concepts. Verification of the query with regards to the ontology information allowed more value to be given to the semantic search in those cases in which the user's query can be completely represented with ontology concepts. This value is decreased when it cannot be completely represented in this way. Once the documents have been combined, they are shown to the user.

6. IMPLEMENTATION AND TESTING

Figure 7 shows a design diagram of the prototype developed to improve document searches in a digital computer science library. In implementation, the programming language Java was used, and some open source tools such as Apache Lucene, Apache Tika, GATE, and Apache Jena were integrated. The prototype is made up of four components. The document processor processes the documents as described in sections 3 and 4. Persistence stores the information obtained during document processing in two repositories: "Lucene file index" and "Semantic annotations." In the first, documents are represented as vectors of terms. The second repository contains the semantic annotations obtained during the annotation process, which are stored as RDF triples. The searching component includes the search functions offered to the user and is made up of three modules: IR search, which processes user queries based on classic information retrieval techniques; semantic search, which processes them based on semantic annotations; and hybrid search, which combines the results of the previous two. Finally, the GUI component allows the user to search for semantically enriched documents using keywords. The user does not need to know the ontology structure or the formal query languages to semantically retrieve documents. The search process is transparent for the user.

Figure 7. Prototype design diagram

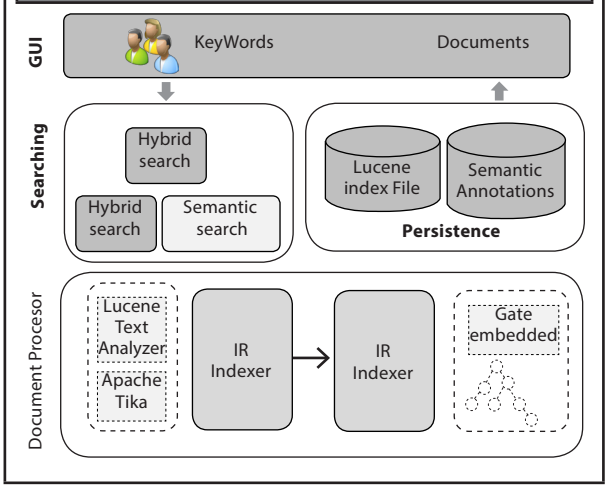


Figure 6. Sparql query generated form keywords

```
SELECT ?Annot  docID  ?Weight WHERE{

 {?Annot   Uv:concept 'Association_rules '  .
   ?Annot  Uv:Weight  ?Weight.
   ?Annot  Uv:doc_id  ?docID .
 }
 UNION
{ ?Annot  Uv:concept 'Data_mining' .
  ?Annot  Uv:Weight  ?Weight.  ?
  ?Annot  Uv:doc_id  ?docID .
}
UNION
{ ?Annot   Uv:ontology_concept  'Mining_methods_and_algorithms '.
  ?Annot   Uv:Weight  ?Weight.
  ?Annot   Uv:doc_id  ?docID .
}
…
ORDER BY DESC(?Weight)
```

The test scenario is made up of a document collection, user queries expressed in keywords, the domain ontology, and semantic annotations. The document collection contains approximately 2,000 computer science documents. The domain ontology has approximately 840 classes. The tests included both queries that could be completely represented with ontology information and queries that could only be partially represented or not represented at all in the ontology. Each query was executed suing three search strategies: IR-based search, semantic-based search, and hybrid-based search. The results obtained with each of these strategies were analyzed based on precision and recall measures. For each query, 10 levels of recall were taken (10%, 20%, 30%,...100%), and for each level of recall, the precision was measured.

Table 4 shows some examples of the queries that were used in the tests. The first three queries correspond to examples in which all the words expressed by the user can be interpreted using the concepts represented in the ontology. In this case the performance of the semantic annotation search was superior to that of the classic search. This is due to the fact that the documents have been enriched by semantic annotations, and these consider the annotation properties and the instances associated to the ontology classes. The final two columns of the table are queries that cannot be completely interpreted with the ontology concepts. Some words, such as "medicine" and "bee pollination," are not represented in the ontology used in the semantic annotation process. In this case, the semantic search failed because there are no annotations related to all the words expressed by the user. In query 4, the semantic search is done considering only the concept "data mining," which affects its performance, as can be seen in **Table 4**.

As can be seen in **Figure 8**, in an average case, IR-based search shows a performance inferior to that of semantic search and hybrid search. For example, in classic search, for recall levels near 0.5, the precision metric also has values of approximately 0.5. From this level on, precision decreases rapidly until reaching zero. On the other hand, in semantic search, for recall values near 0.5, precision values of nearly 0.8 were obtained. Also, semantic search obtains better recall levels because queries are expanded through the properties of annotation and ontology class instances. For its part, hybrid search shows better performance than semantic search on the average. Semantic search works very well when the user expresses queries that can be completely interpreted with the information represented in the ontology. However, its performance is poorer when there are no semantic annotations that allow for a user's query to be interpreted. Hybrid search functions better since it works with the advantages of classic search and semantic search.

This is achieved because the hybrid ranking is calculated depending on the query conditions, as described in phase 3 of section 5.

7. CONCLUSIONS AND FUTURE WORK

This study developed a prototype for a semantically extended information retrieval system. This system shows better performance in terms of precision and recall than a system based solely on classic IR techniques. The semantic annotations and domain ontology are a fundamental part of this system. When user queries can be completely interpreted with the ontology concepts, the semantic annotation search offers better results than the classic search. However, if the ontology is incomplete, the semantic search can fail because the annotations do not cover all the semantics of the documents. Therefore, this proposal is based on the hybrid search paradigm that works with the advantages of classic information retrieval and semantic annotation retrieval. Other proposals, like those of Rodríguez-García et al. (2014b) and Bikakis (2010) are only based on semantic annotations that are obtained based on ontologies.

In terms of precision and recall, a semantically extended information retrieval system offers better results than a system based solely on the application of classic IR techniques. The precision measurement improved because the document search considers ontology concepts, which have an explicitly defined meaning. The recall measurement improved because the queries are expanded through instances and annotation properties associated with the ontology classes. The semantically extended system retrieves documents not only with the words expressed by the user, but also extends the concepts that are semantically related in the ontology.

Figure 8. Precision vs. recall for case averages

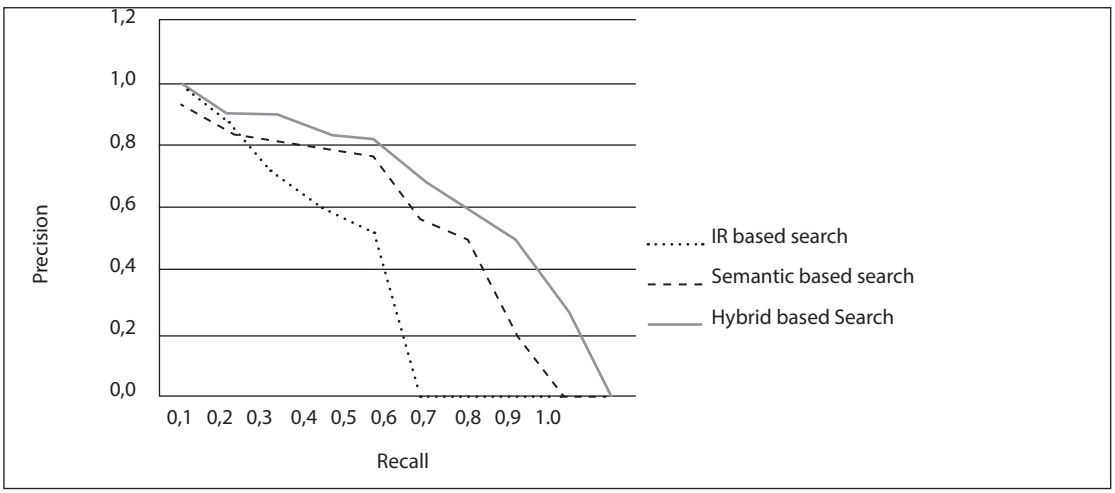


Table 4. Results of some queries in terms of precision (P) and recall (R)

#	Query	Classical IR		Semantic		Hybrid	
		P	R	P	R	P	R
1	Association rules in data mining	0,54	0,59	0,81	0,9	0,72	0,84
2	Cost estimation in software engineering	0,57	0,61	0,77	0,89	0,70	0,84
3	Operating systems	0,60	0,65	0,87	0,92	0,75	0,81
4	Data mining applied to medicine	0,57	0,63	0,35	0,43	0,59	0,63
5	Artificial bee pollination algorithms	0,54	0,61	0,30	0,41	0,59	0,67

Ontologies help to improve the results obtained in information retrieval systems. These are generally implemented in OWL or RDF and are queried using languages like Sparql. Since in this proposal the user expressed queries using keywords, it was necessary to create an index of concepts to relate a user query with the ontology classes. The concept index allows users to access semantic annotations and information represented in the ontology without having to know query languages like Sparql. This proposal differs from studies like those of Castells et al. (2007) in which the query is expressed using Sparql, which can represent a high level of complexity for users.

This proposal offers the possibility of semantically searching for documents without the user having to know the ontology structure that was used during the annotation process. Information retrieval is transparent for the user, who must only express a set of keywords. The system automatically selects the ontology classes that will orient the search process. In this sense, this proposal differs from others like those of Bhagdev et al. (2008) and Bikakis et al. (2013) in which the user must know the ontology structure and manually select the classes that will be used in the document search.

In the development of this study, we integrated tools from the field of information retrieval and also from the Semantic Web. The integration of these fields of knowledge offers great advantages for improving the effectiveness and performance of document retrieval systems. The tests were carried out with a computer science ontology. However, the prototype allows for configuring an ontology from any domain. The improvement of results depends on the quality and completeness of the information represented in the ontology.

Future work

As future work, we propose including multiple domain ontologies in the annotation and document retrieval processes, as well as working with ontologies that offer more relationships and hierarchies in order to be able to support more complex user queries. It is also necessary to continue exploring techniques that allow the user to access information stored in the ontologies in a usable and natural way without having to know the formal query languages. Another aspect is related to the scalability of semantic annotation search tools.

We must look for mechanisms that allow users to access semantic annotations in large-scale environments with minimum response times. This requirement was not evaluated in this study, but it must be considered in order to implement a system in which many simultaneous users can access semantic annotations.

REFERENCIAS

Baeza, R.; Ribeiro, B.A. (1999). Modern Information Retrieval. ACM Press/New York, Addison-Wesley.

Bikakis, N.; Giannopoulos, G.; Dalamagas, T.; Sellis, T. (2010). Integrating keywords and semantics on document annotation and search. *Springer-Verlag* Berlin, Heidelberg, pp.921-938.

Bhagdev, R.; Chapman, S.; Ciravegna, F.; Lanfranchi, V.; Petrelli, D. (2008). Hybrid Search: Effectively Combining keywords and Semantics Searches. *Springer Berlin* Heidelberg, LNCS 5021, pp. 554–568.

Bontcheva, K.; Cunningham, H.; Kiryakov, A.; Tablan, V. (2006). Semantic Annotation and Human Language Technology. Semantic Web Technologies: Trends and Research in Ontology-based Systems, Davies, J.;Studer, R.; Warren, P. John Wiley & Sons, Ltd, pp. 29-50.

Castells, P.; Fernández, M.; Vallet, D. (2007). An Adaptation of the Vector-Space Model for Ontology-Based Information Retrieval. *IEEE Transactions on Knowledge and data Engineering*, 19(2), pp. 261-272.

Corcho, O. (2006). Ontology based document annotation: trends and open research problems. *Inderscience Publishers*, 1(1), pp. 47–57.

Kiryakov, A.; Popov, B.; Ognyanoff, D.; Manov, D.; Terziev, I. (2004). Semantic annotation, indexing, and retrieval. J*ournal of Web Semantics: Science, Services and Agents on the World Wide Web*, 2(1), pp. 49-79.

Lee, T., Hendler, J., Lassila, O. (2001). The semantic Web, *Scientific American*, 284(5), pp. 34-43.

Lei, Y.; Uren, V.; Motta, E. (2006). Semsearch: A search engine for the semantic Web. *Springer Berlin* Heidelberg, 4248, pp.238-245.

Mangold, C. (2007). A survey and classification of semantic search approaches. *International Journal of Metadata, Semantics and Ontologies*, 2(1), pp.23-34.

Manning, C.D.; Raghavan, P.; Schütze, H. (2008). Introduction to Information Retrieval. Cambridge University Press.

Nagypal, G. (2005). Possibly imperfect ontologies for effective information retrieval. PhD thesis, University of Karlsruhe, 3762, pp.780-789.

Nesić S.; Jazayeri M.; Crestani, F.; Gašević, D. (2010). Concept-Based Semantic Annotation Indexing and Retrieval of Document Units. In *Proceedings of the 9th International conference on Adaptivity, Personalization and Fusion of Heterogeneous Information*. Paris, France, RIAO, pp.234–237.

Oren E.; Moller K.; Scerri S.; Handschuh S.; Sintek M. (2006). What are semantic annotations. Technical report, DERI Galway .

Popov, B.; Kiryakov, A.; Ognyanoff, D.; Manov, D.; Kirilov A.; Goranov, M. (2004). KIM semantic platform for information extraction and retrieval Journal . *Natural Language Engineering*, 10(3-4), pp. 375-392.

Porter M.F. (1997). An algorithm for suffix stripping. In Readings in information retrieval, Sparck, K. and Willett, P. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA 313-316.

Rodríguez, M.A.; Valencia, R.; García, F.; Samper, J.J. (2014). Creating a semantically-enhanced cloud services environment through ontology evolution. *Future Generations in Computer Systems*, 32, pp. 295–306.

Rodríguez, M.A.; Valencia, R.; García, F.; Samper, J.J. (2014). Ontology-based annotation and retrieval of services in the Cloud. *Knowledge-Based Systems*, 56, pp. 15-25.

Salton, G.; Wong, A.; Yang, C.S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18(11), pp. 613–620.

Samper, J.J.; Adell, F.J.; Van den Berg, L.; Martinez, J.J. (2008). Improving Semantic Web Service Discovery. *Journal of Networks* (JNW), 3(1), pp.35-42.

Shah, U.; Finin, T.; Joshi, A.; Scott Cost, R.; Matfield, J. (2002). Information retrieval on the semantic Web. *International conference on Information and knowledge management*, New York, pp. 461-468.

Strasunskas, D.; Tomassen, S. (2010). On Variety of Semantic Search Systems and Their Evaluation Methods. International Conference on Information Management and Evaluation. South Africa. A*cademic Conferences Publishing*, pp. 380-387.

Tan P.N.; Steinbach M.; Kumar, V. Introduction to Data Mining. Addison-Wesley, chapter 2, pp 74.

Tran, T.; Wang, H.; Rudolph, S.; Cimiano, P. (2009). Top-k Exploration of Query Candidates for Efficient Keyword Search on Graph-Shaped (RDF). *Data Engineering Conference, IEEE 25th International*, pp. 405-416.

TrivikRam, I., (2007). A Hibrid Approach to retrieving Web documents and Semantic data. Phd. tesis Wright State University, pp. 30.

Vallet, D.; Fernández, M.; Castells, P. (2005). An Ontology-Based Information Retrieval Model. *The Semantic Web: Research and Applications. Springer Berlin* / Heidelberg, 3532, pp. 455-470.

Wang, H.; Zhang, K.; Liu, Q.; Tran, T.; Yu, Y. (2008). Q2semantic: A lightweight keyword interface to semantic search. *The Semantic Web: Research and Applications. Springer* Berlin Heidelberg, 5021, pp. 584–598.

Wei, W.; Barnaghi, P.M.; Bargiela, A. (2007). Semantic-Enhanced Information search and Retrieval. *Conference on Advanced Language Processing and Web Information Technology,* Luoyang, Henan, China, pp. 218-223.

Wei, W.; Barnaghi, P.M..; Bargiela, A. (2008). Search with Meanings: An Overview of Semantic Search Systems. *Inter. Journal of Communications of SIWN*, 3, pp. 76-82.

Uren, V.; Lei, Y.; López, V.; Liu., H.; Motta, E. (2007). The usability of semantic search tools: a review. *The Knowledge Engineering Review*, 22(4), pp.361-377.

Zhou, Q.; Wang, H.; Xiong, M.; Wang C.; Yu, Y. (2007). SPARK: adapting keyword query to semantic search. In *Proceedings of the 6th international The semantic Web. Springer-Verlag*, Berlin, Heidelberg, 4825, pp. 694-707.

TO REFERENCE THIS ARTICLE / PARA CITAR ESTE ARTÍCULO / PARA CITAR ESTE ARTIGO /

Solarte-Pabón, O.; Millán, M. (2014). Proposal for semantically extending the information retrieval process. Revista EIA, 11(22) July-December, pp. 45-58. [Online]. Available on: http://dx.doi.org/10.14508/reia.2014.11.22.49-63 reia.2014.11.22.51-65