TWebS: An Application of Terminological Logics in Web Searching

Alvaro Barreiro [1], DavidE.Losada[1] and RaúlRamos[1,2]

[1] Dept.Computación.Fac.Informática
Universidad de A Coruña.
Campus de Elviña 15071 A Coruña, Spain
{ barreiro, losada }@dc.fi.udc.es

[2] CERN.EuropeanLaboratoryforParticlePhysics
Information Technology Division
1211 Genève, Switzerland
Raul.Ramos@cern.ch

Keywords: Intelligent Information Retrieval, Searching the World-Wide Web, Terminological Systems, Description Logics.

1. Introduction

Classical techniques of **Information Retrieval** (IR) [1] have been successfully applied in the design of Internet search tools. Tools such as *Yahoo!, Lycos, Altavista or Olé*, provide simple and efficient search methods, mostly based on indexes and syntactic search. However, in many cases simple indexing is not enough. As an example, consider you are using a simple indexing search engine (*robot*) to find documents about Mathematics. Next, you look for documents about Algebra. You will surely discover that many Algebra documents were not found during the first search. In order to get these documents in the former search, the robot should make use of semantic information, i.e. the fact that Algebra documents are Mathematics documents. Obviously, this knowledge can not be represented in a conventional robot based only on lexical indexing. We believe that Internet users, although not entirely aware of the above analysis, experience the advantages and limitations of the simple indexing approach to Web searching. In fact, users often look semantically into the documents provided by the first lexical search to check if their expectations are fulfilled. Moreover, robots based on standard IR boolean models do not capture well the hypertext structure of the data.

In this paper we present **TWebS** (Terminological Web Searching), a system that makes use of semantic information when searching the Web. Our approach is in the framework of **Knowledge Representation** (KR) and particularly in the field of **Description Logics** (DLs). User expectations are expressed in a Knowledge Base (KB) and TWebS accesses the Web and fills the KB. This way, beginning from a KB and a website, TWebS produces a set of assertional axioms extracted from HTML documents and relevant to the user's expectations. DLs are a suitable formalism to build TWebs because: (1) they clearly distinguish between intensional (ontological) and extensional (object specific) knowledge, (2) the same language can be used to express queries, (3) document modelling from content, structure and other perspectives can be accomplished and (4) DL descriptions are suitable for representing the semi-structured file format.

In TWebS the model for documents is split in two definitional modules: the User-Tbox and the HTML-TBox. The User-TBox (UTBox) models the user's view of the documents. The UTBox basically contains his/her expectations faced with the task of exploring the Web. The HTML-TBox (HTBox) models the semi-structured HTML format. The necessity of these efforts to formalize (in a KR sense) the semantics of HTML tags has been emphasized [2]. The division in two definitional modules has several advantages: (1) expressing the HTBox as a separate module makes TWebS modularly extensible to any semi-structured format, (2) facilitates system design because is in HTBox where the patterns that must be searched in the documents are clearly specified, avoiding hand-coded programming.

The rest of this paper is organised as follows. Section 2 briefly explains DLs and terminological models of IR [3]. Section 3 describes our approach for extracting assertional axioms from the Web starting from a model expressed as a terminological KB. Section 4 covers the design and some implementation choices of TWebS. Finally, in section 5 we conclude with a discussion of advantages and disadvantages of our approach.

2. Description Logics and the Terminological Model of Information Retrieval

Description Logics are a family of logics devised to represent, organise and manipulate knowledge of a particular application domain (a terminology) by means of taxonomies of concepts and roles. DLs are considered as formal successors of semantic networks and more specifically of KL-ONE [4]. After the pioneering work of Levesque and Brachman [5], the compromise between expressiveness of the languages and decidable and tractable reasoning has been the object of theoretical studies in the last decade yielding important results. A compendium of complexity results can be found in [6] and [7] is a general study relating the expressiveness of DLs with Predicate Calculus. From an applied perspective, research has been done in the implementation of Concept Languages (CLs), among them CLASSIC [8] is a well known tractable language [9], and practical applications of these CLs. In particular, there are applications for semantic data modelling [10] and proposals of terminological models of IR [2,11].

DLs see the world as a set of objects, here called *individuals*. *Concepts* denote subsets of individuals and *roles* denote binary relations between individuals. Concept languages provide term constructors for building variable-free composite terms that associate concept and roles to define new concepts and roles. The TBox (Terminological Box) or Definitional Module contains the intensional knowledge which denotes global knowledge about a specific domain. A taxonomy composed of concepts and roles and their subsumption relationships is maintained in the TBox. The Abox (Assertional Box) or Assertional Module contains the set of individuals and relations between them that give rise to a specific perception of the world.

Next, we formally introduce the basic notions of CLs. A brief exposition of the terminological model of IR, in whose framework our system is accommodated, concludes the section. We will restrict ourselves to the constructors of the CLASSIC language.

2.1. Syntax and Semantics

Let **A** be a set of atomic concepts and **P** a set of atomic roles. Concepts (C, D) and roles R are inductively built from atomic concepts and roles, the universal concept and individuals i, using the language constructors: (1) An element of **A**, A, is a concept and an element of **P**, P, is a role (atomic concepts and roles), (2)$\top$(universal concept), (3) $C \sqcap D$ (intersection), (4) $\forall R.C$ (universal role quantification), (5) $\geq nR$, $\leq nR$ (number restrictions), (6) *fills* R i (fills) and (7) $f_1 \circ f_2 \circ \ldots \circ f_k = g_1 \circ g_2 \circ \ldots \circ g_h$ (equality restriction for attributes). Notice that in this language there are only atomic roles and attributes since it does not have role constructors.
The formal meaning of the language is given by a model-theoretic interpretation $\mathcal{I} = (\mathcal{D}^{\mathcal{I}}, .^{\mathcal{I}})$. The interpretation consists of an arbitrary set $\mathcal{D}^{\mathcal{I}}$ (the domain of the interpretation) and an interpretation function $.^{\mathcal{I}}$ that maps every concept A in a subset of $\mathcal{D}^{\mathcal{I}}$ and every role P in a subset of $\mathcal{D}^{\mathcal{I}} \times \mathcal{D}^{\mathcal{I}}$ ($P^{\mathcal{I}}$). The predefined concept $\top$ has a fixed interpretation, $\mathcal{D}^{\mathcal{I}}$. The meaning of the concept expressions is:

$(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$
$(\forall R.C)^{\mathcal{I}} = \{a \in \mathcal{D}^{\mathcal{I}} \mid \forall\ \mathrm{b.(a,b)} \in R^{\mathcal{I}} \longrightarrow \mathrm{b} \in C^{\mathcal{I}}\}$
$(\geq nR)^{\mathcal{I}} = \{a \in \mathcal{D}^{\mathcal{I}} \mid \mathrm{card}\{\ \mathrm{b} \mid (a,b) \in R^{\mathcal{I}}\} \geq \mathrm{n}\ \}$
$(\leq nR)^{\mathcal{I}} = \{a \in \mathcal{D}^{\mathcal{I}} \mid \mathrm{card}\{\ \mathrm{b} \mid (a,b) \in R^{\mathcal{I}}\} \leq \mathrm{n}\ \}$
$(fillsRi)^{\mathcal{I}} = \{a \in \mathcal{D}^{\mathcal{I}} \mid \mathrm{i} \in \mathcal{D}^{\mathcal{I}} \wedge\ \mathrm{(a,i)} \in R^{\mathcal{I}}\}$

Attributes are also included in the language. Attributes are functional roles and must be interpreted as partial functions rather than arbitrary binary relations. Therefore, the meaning of the equality restriction is given by:

$$(f_1 \circ f_2 \circ \ldots \circ f_k = g_1 \circ g_2 \circ \ldots \circ g_h)^{\mathcal{I}} = \{a \in \mathcal{D}^{\mathcal{I}} \mid \mathrm{f}_k{}^{\mathcal{I}}(\ldots \mathrm{f}_1{}^{\mathcal{I}}(\mathrm{a})) = \mathrm{g}_h{}^{\mathcal{I}}(\ldots \mathrm{g}_1{}^{\mathcal{I}}(\mathrm{a}))\}$$

2.2.Tbox and ABox

For the formal introduction of terminologies we will follow the notation of [12].Let D be a concept expression and S a role expression. A TBox or *Terminology* is a finite set of *terminological axioms* that define the concepts A, B and role R: (1) Terminological axioms of defined concepts and roles (also called complete definitions): $A = D$, $R = S$, (2) Terminological axioms of primitive concepts and roles (also called incomplete definitions): $A \sqsubseteq D, R \sqsubseteq S$ and (3) Terminological disjointness axioms: *dis(A,B)*. Axioms must satisfy two restrictions: (1) a concept or role cannot appear more than once on the left hand side of a terminological axiom, and (2) the disjointness axiom must not contain defined concepts.
Let A be a concept, R a role, D a concept expression and S a role expression. An interpretation $\mathcal{I} = (\mathcal{D}^{\mathcal{I}}, .^{\mathcal{I}})$ satisfies a terminological axiom iff: $A^{\mathcal{I}} = D^{\mathcal{I}}(R^{\mathcal{I}} = S^{\mathcal{I}})$ for the terminological axiom *A=D (R=S)*, $A^{\mathcal{I}} \subseteq D^{\mathcal{I}}(R^{\mathcal{I}} \subseteq S^{\mathcal{I}})$ for the terminological axiom $A \sqsubseteq D(R \sqsubseteq S)$ or $A^{\mathcal{I}} \cap B^{\mathcal{I}} = \emptyset$ for the terminological axiom *dis(A,B)*.
An ABox contains all the individuals and relations that are part of the world defined in the TBox. If C is a name of concept, R is a name of role, and a and b are names of individuals then *(a.C)* and *(a.b.R)* are assertional axioms.

The interpretation function $.^{\mathcal{I}}$ is extended to map the names of the individuals over $\mathcal{D}^{\mathcal{I}}$. An interpretation $\mathcal{I}$ satisfies an assertional axiom iff: $a^{\mathcal{I}} \in C^{\mathcal{I}}$ for the assertional axiom *(a.C)* and $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$ for the assertional axiom *(a.b.R).*
Now we can define a *model.* An interpretation is a model of a TBox if it satisfies all the terminological axioms and an interpretation is a model for an ABox if it satisfies all the assertional axioms. Terminological systems usually include the *unique name assumption* but do not include the *closed world assumption.*

2.3. Basic Inferences and Rules

Subsumption is the more general inference in CLs and satisfiability, equivalence and disjointness problems can be reduced to subsumption problems [6]. Formally, a concept C is subsumed by a concept D, $C \sqsubseteq D$, *if* $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for each interpretation $\mathcal{I}$. CLASSIC has also a simple forward-chaining inference mechanism. A CLASSIC rule consists of an antecedent concept and a consequent concept, where the antecedent must be a defined concept. When and individual is known to satisfy the antecedent concept, the rule is *triggered* and the individual is also known to satisfy the consequent. With this mechanism, descriptions can be attached to concepts as rule consequents. Rules are useful to introduce non-definitional aspects of the antecedent concept. In the following we will denote rules as $A \Rightarrow C$, where A is the name of a concept and C is a concept description.

2.4. Terminological Model of Information Retrieval

Logical models view the task of IR as the extraction from a document base and given a query q, of those documents d that satisfy $d \rightarrow q$, where d and q are well formed formulae of the logic and $\rightarrow$ is the logical implication. If DLs are the chosen logic, the terminological model of IR is obtained: the task of IR becomes that of extracting those documents d such that $d \sqsubseteq q$ where d and q are *terms* of the chosen DL and $\sqsubseteq$ is the *subsumption* relation.
Besides the selection of a DL, the terminological model of IR [2] also endows IR systems with representations of the documents, queries and lexical knowledge. Therefore, the terminological model is composed of: (1) **A model for documents**. A document is an individual of the logic and assertional axioms referring to it constitute descriptions of the documents. Documents can be multifaceted modelled. Adopting a DL as the modelling language, contextual attributes, internal structure, physical appearance (layout) and semantic content can be addressed in the same formalism. *Incrementality* is also an advantage of the terminological modelling. (2) **A model for queries**. A query is a concept or role expression and represents the individuals, or pairs of individuals, which satisfy the expression. (3) **A model for lexical entries**. A set of terminological axioms that allow the specification of the meaning of the predicate symbols used in both document and query models.

3.Splitting the model for documents

Given a model for the documents expressed as a TBox and a document base obtained from the Web, TWebS automatically generates assertional axioms that describe the documents and loads the asserts in the ABox. The generic task of IR can be carried out with the reasoning mechanisms of a terminological system.

The model for documents is split in two definitional modules: the User-TBox and the HTML-TBox. The User-TBox (UTBox) models the user's view of the documents. The UTBox basically contains his/her expectations faced with the task of exploring the Web. Thesaurus knowledge can also be included in this module in order to improve the search results. The HTML-TBox (HTBox) models the semi-structured HTML format. Both of them can represent documents from the multiperspective view (contextual, structure, layout and content) which is a trademark of the terminological model of IR.
The HTBox is a terminological description of the HTML document format. Expressing it as a separated module makes TWebS modularly extensible to other semi-structured formats. The key idea is to conceptualise the HTML *tags*. This way the HTBox becomes a definitional module of conceptualised roles and attributes. This allows us to take advantage of the mechanisms of terminological systems: we can define, incrementally define, specialise, generalise, classify or instantiate HTML tags.
Next, we show a simple HTBox which defines three HTML tags: TITLE, H1 and KEYWORDS.

STRING $\sqsubseteq \top$
INTEGER $\sqsubseteq \top$
pattern $\sqsubseteq \top \times \top$
value $\sqsubseteq \top \times \top$
TAGS = (($\forall$ pattern.STRING) $\sqcap$ (= 1.pattern))
ROLES = (TAGS $\sqcap$ ($\geq$ 1.value))
ATTRIBUTES = (TAGS $\sqcap$ ($\geq$ 1.value) $\sqcap$ ($\leq$ 1.value))
ATTRIBUTESS = (ATTRIBUTES $\sqcap$ ($\forall$ value.STRING))
ATTRIBUTESI = (ATTRIBUTES $\sqcap$ ($\forall$ value.INTEGER))
ROLESS = (ROLES $\sqcap$ ($\forall$ value.STRING))
ROLESI = (ROLES $\sqcap$ ($\forall$ value.INTEGER))
TITLE = (ATTRIBUTESS $\sqcap$ (fills pattern "TITLE"))
H1 = (ROLESS $\sqcap$ (fills pattern "H1"))
KEYWORDS = (ROLESS
$\sqcap$ (fills pattern "META NAME=KEYWORDS CONTENT '"))

CLs usually have *host* concepts such as STRING and INTEGER. The role pattern relates each tag with its pattern and the role value represents the binary relation between a tag and its value. The concept TAGS conceptualises HTML tags: i.e. the set of individuals which have exactly one pattern and the pattern is a string. ROLES is a specialisation of TAGS. Notice that someone could think that ROLES = TAGS would be a better definition but this is put in charge of the model builder. He/she does not seem to be interested in the tags that have no values. The definitions construct a taxonomy of further specialisations and the three last terminological axioms define the HTML tags in which we were interested. In what follows, we are assuming that the HTML tag KEYWORDS could be useful for a keyword based document representation. Other assumptions closer to the traditional techniques of IR could have been done.
The UTBox is the actual model of the documents from the user's viewpoint. In the following example thesaurus information has been included in the UTBox assuming that the user is interested in certain relations.

STRING $\sqsubseteq \top$
INTEGER $\sqsubseteq \top$
title $\sqsubseteq \top \times \top$
h1 $\sqsubseteq \top \times \top$
keywords $\sqsubseteq \top \times \top$
MATDOC $\sqsubseteq \top$
ALDOC $\sqsubseteq$ MATDOC
CALDOC $\sqsubseteq$ MATDOC
hasMAT = fills keywords "Mathematics"
hasAL = fills keywords "Algebra"
hasCAL = fills keywords "Calculus"
EXDOC = MATDOC $\sqcap$ (fills title "Examination")
EXEXDOC = EXDOC $\sqcap$ (fills h1 "Exercises")
hasMAT $\Rightarrow$ MATDOC
hasAL $\Rightarrow$ ALDOC
hasCAL $\Rightarrow$ CALDOC

Axioms 6 to 8 construct a taxonomy where the thesaurus information is saved. The user is interested in Mathematics documents in general, Algebra documents and Calculus documents. The individuals having a certain filler for its role keywords are classified in the thesaurus taxonomy via the three rules in the module. Axioms 12, 13 of the UTBox defines two concepts by means of its fillers for the roles title and h1. The concept EXDOC has the individuals containing the title Examination and dealing with Mathematics. A document with Title Examination, keywords Calculus or Algebra, and where Mathematics does not belong to its keywords, is contained in EXDOC. Therefore the thesaurus information is very useful to improve search results.

3.1.Assertional Axioms

TWebS produces assertional axioms about the HTML tags that, at the same time, are conceptualised in the HTBox and are used in the UTBox. The former condition guarantees that the tag can be found when TWebS searches the text of HTML documents. If the second condition does not hold, the tag in question does not concern TWebS.
Consider the two definitional modules presented above, and a website that provide two documents, Html1 and Html2. Let us suppose that Html1 contains one tag TITLE and two tags KEYWORDS containing respectively “Examination”, “Algebra” and “Homomorfism”. Let us suppose that Html2 contains a tag TITLE and a tag H1 containing respectively “Informatics” and “Compilers”. Then, TWebS produces the following assertional axioms:

(Html1."Examination".title)
(Html1."Algebra".keywords)
(Html1."Homomorfism".keywords)
(Html2."Informatics.title)
(Html2."Compilers".h1)

If the user asks about the instances of the concept MATDOC, the document Html1 will be retrieved, although the string “Mathematics” does not appear in the keywords of the document.

3.2. TWebS and IR Boolean Model

In the model of documents presented, documents can be defined using a description involving a set of roles whose fillers are values of HTML tags. The classical Boolean Model, where a document is a set of terms from a fixed set, can also be mapped into TWebS. In the UTBox a document can be defined via a description involving a role and restricting the set of its possible fillers. However CLASSIC has only a restricted treatment of disjunction and negation.

4.Design and Implementation of TWebS

In TWebS the Web Access and Exploration Module (WAEM) produces the document base given the URL corresponding to a website. The WAEM is composed of two submodules: the Web Access Module (WAM) and the Exploration Module (EM).
The WAM obtains individually an HTML document from its URL. The EM is actually the agent which constructs the document base. It takes the URL of a website as input and uses the WAM to retrieve the HTML document corresponding to that URL. Then, it searches the HTML document for links (URLs) to other HTML documents, which are recursively explored if the referred document fulfills the following conditions: (1) it has not been explored yet (cycles are avoided) and (2) it lives in the same server as the referring document.
The ABox Generation Module (AGM) produces the Assertional Module associated to the document base it takes as input. The Assertional Axiom Generation Module (AAGM) and the Template Generation Module (TGM) are AGM's submodules. The AAGM uses a template to construct the axioms and explores each document using the information contained in the template, which indicates what to look for and how to generate the axioms. The Template Generation Module (TGM) builds the template the AAGM needs. For this, it interacts with the terminological system, coordinating the information in the UTBox and the HTBox to obtain the list of roles which can be involved in assertions. As seen above, TWebS can only generate assertional axioms about the roles that are used by the user in the UTBox and are conceptualised in the HTBox. Therefore, the TGM makes an entry in the template for every role that fulfills these conditions. Basically, every entry contains a generic axiom (this is a template) to be generated by the AAGM about the documents, and the TGM fills the entry with all the information needed to construct the assertional axiom. This information is obtained from the HTBox. Figure 1 shows the structure of an entry of the template.

Tag	Pattern	Individual_1	Individual_2	Role	Axiom	Attribute/Role

Fig. 1. Structure of the template to generate assertional axioms

The Tag column contains the name of the conceptualised tag in the HTBox. The Pattern column contains the string that the system tries to find into the documents. The Axiom column is the pattern to create the axioms based on information in the Role, Individual1 and Individual2 columns. The Attribute/Role column is a boolean which indicates

whether the role is functional. Therefore, the system knows if it has to continue looking for more instances of that tag into the HTML document. The Individual1, Individual2 and Role columns contain the role and associated individuals needed to fill the axioms.
We have selected CLASSIC as the language to build the terminological system. AT&T Bell Laboratories make CLASSIC available to academic researchers and it runs in several platforms.CLASSIC has been used in a great number of applications in commercial and prototypical form. Particularly, we use NeoClassic that is the newest version of CLASSIC. NeoClassic is written in C++ and its facilities are available from C++ code through a fully documented API (Application Programming Interface). The latter reason was very important for us in order to integrate TWebS into a modular environment with communications with other systems.
The AGM needs to communicate with CLASSIC to perform several tasks. The submodule TGM needs to obtain the set of roles that the user defines in the UTBox and to determine if a role in the UTBox is conceptualised in the HTBox. The AAGM needs to load the ABox on the Terminological System. All the interaction between the AGM and the Terminological System was made using the API provided by CLASSIC.

Acknowledgements : This work was supported in part by project 10503B96 from Xunta de Galicia.

References

[1] Frakes, W. B. and Baeza-Yates, R. Information Retrieval: Data Structures and Algorithms. Prentice Hall Inc. NJ, 1992.
[2] Welty, C. DLs for DLs : Description Logics for Digital Libraries. Proc. of the 1998 Int. Workshop for Description Logics, Trento, Italia, 1998.
[3] Meghini, C. and Sebastiani, F. and Straccia, U. and Thanos, C. A model of information retrieval based on a terminological logic. "Proc. of SIGIR-93 ACM Conference on Research and Development in Information Retrieval", 1993, 298-307.
[4] Brachman, R. J. and Schmolze, J. G. An overview of the KL-ONE knowledge representation system. Cognitive Science, 9(2), 1985, 171-216.
[5] Levesque, H. J. and Brachman, R. J. A fundamental tradeoff in knowledge representation and reasoning (revised version). Readings in Knowledge Representation. Morgan Kauffman, Los Altos, CA, 1985, 817-823.
[6] Donini, F. and Lenzerini, M. and Nardi, D. and Nutt W. The complexity of concept languages. Information and Computation, 134(1), 1997, 1-58.
[7] Borgida, A. On the relative expresiveness of description logics and predicate logics. Artificial Intelligence, 82, 1995, 353-367.
[8] Borgida, A. and Brachman, R.J. and McGuiness, D.L. and Resnick, L.A. CLASSIC: A structural data model for objects. "Proc. ACM SIGMOD Conference on Management of Data", 1, 1989, 59-67.
[9] Borgida, A. and Patel-Schneider, P. A semantic and complete algorithm for subsumption in the CLASSIC description logic. "Journal of Artificial Intelligence Research", 1, 1994, 277-308.
[10] Borgida, A. Description logics in data management. "IEEE Transactions on Knowledge and Data Engineering", 7(5), 1995, 671-682.

[11] Meghini, C. and Straccia, U. A relevance terminological logic for information retrieval. "Proc. of SIGIR-96 ACM Conference on Research and Development in Information Retrieval", 1996.
[12] Baader, F. A formal definition for the expressive power of terminological knowledge representation languages. "Journal of Logic and Computation", 6(1), 1996, 33-54.
[13] Catarci, T. and Iocchi, L. and Nardi, D. and Santucci, G. Conceptual views over the Web. "Proc. of KRDB Workshop", 1997