Improving Web Information Retrieval using Ontologies: an Extension and Implementation of OWL for Web Queries Enrichment

Nicolas Guelfi
LASSY – University of Luxembourg
Nicolas.guelfi@uni.lu

Cédric Pruski
LASSY – University of Luxembourg
and
LRI-Univ. of Paris-Sud-CNRS & INRIA-Futurs,
Cedric.pruski@uni.lu

Chantal Reynaud
LRI-Univ. of Paris-Sud-CNRS & INRIA-Futurs
Chantal.reynaud@lri.fr

Abstract

The use of Semantic Web technology comes into general use these last years. This is all the more true concerning ontology languages. Moreover, since the standardization of the Web Ontology Language, ontologies designed using OWL are becoming numerous over the Web. Therefore, applications are needed to benefit from this phenomenon. In this paper, we will illustrate how OWL ontologies can improve Web information retrieval. First of all, we present the adaptation of existing OWL primitives to our "Optimal ontology-based Web Information Retrieval" (O^3) approach. We also discourse the integration of three new semantic relations into OWL in order to enrich O^3. Subsequently, the definition of rigorous query expansion rules and consequential experimental results obtained using the TARGET prototype allows us to emphasize the contribution of OWL ontologies in our approach for improving Web documents' relevance when searching the Web.

1. Introduction

Since the advent of the Internet in the early nineties, the increasing number of pages constituting the Web requires the development of adapted tools in order to assist users to retrieve information in a relevant way. This is also one of the objectives of the Semantic Web [3]. Its main purpose is to give a sense to the Web using ontologies which will significantly facilitate Web documents retrieval.

In our previous work [7], we have proposed the O^3 general approach. It uses the WordNet [5] linguistic tool in order to optimize, in terms of relevance, the returned documents when searching the Web. Its main idea consists in enriching, following well-defined rules, the query constructed by users by extracting from WordNet the appropriate vocabulary that characterizes best the search domain. O^3 has been formalized using first-order logic and graph theory. This formal framework permitted the rigorous definition of query expansion rules. In parallel, the standardization of OWL [11] has hastened the quick and massive development of OWL ontologies across the Web. This is why, to benefit from both O^3 and OWL ontologies, we decided to make O^3 compatible with OWL. Therefore, we had first to find equivalence between existing OWL primitives and the linguistic relations of WordNet and second, define new OWL primitives to express meronymy and antonymy. Our goal is to show how O^3 would benefit from an extension of OWL. This improvement turns up through the newly defined query enrichment rules, implemented in the TARGET tool, which has shown to be extremely effective. Nevertheless, since we aim at improving Web search, our long-term objective is to integrate more linguistic relations into OWL. These relations will be identified and formalized after a deep study of Web evolution.

In this paper, we propose to study the possibilities offered by OWL that cope with O^3 as well as an extension of the language. We also present an implementation of the so extended OWL through query enrichment rules and an experimental validation using the TARGET system. The remainder of this paper is organized as follows: section 2 presents both the O^3 approach and the extension of OWL. Section 3 deals with query expansion starting with state-of-the-art. Section 4 introduces the experimental results. Finally, last section wraps up with our concluding remarks and future work.

2. Standard OWL and extended OWL

The O^3 approach implements the following semantic relations borrowed from WordNet: synonymy (relation between terms), meronymy, hyperonymy and antonymy (relations between synsets). Our first task was to define OWL equivalence for these relations. On one hand, synonymy, which is a relation between terms, is not represented in OWL, but because of its use in O^3, we exploit the equivalence relation which concerns classes in OWL. Hyperonymy is equivalent to subsumption in OWL. On the other hand, meronymy and antonymy have no equivalence in OWL; therefore, we needed to extend the language in order to be compatible with O^3. In this section, we first present the O^3 approach, then OWL primitives directly exploitable in O^3. The extension of OWL that gives place to a new version of O^3, called O^4, ends the section.

2.1. The O^3 approach

The O^3 approach [7], depicted figure 1, aims at improving, in terms of relevance, the results of a Web search. The first version of the TARGET tool that implements this approach, has given interesting results. It uses Google to interact with the Web and our query language: ASK. A first non-enriched query is submitted to Google which extracts a bunch of Web pages. These documents are transformed, using an ontology of the involved application domain, into WPGraphs and W^{3}Graphs. Parallel to this, the initial query is enriched according to the vocabulary of the same ontology and enrichment rules. Finally, the enriched query is verified on the graphs in order to extract the most relevant documents. The returned information as a result of the query expansion rules, corresponds better to users' expectations. Actually, the rules are designed to target even more the research space.

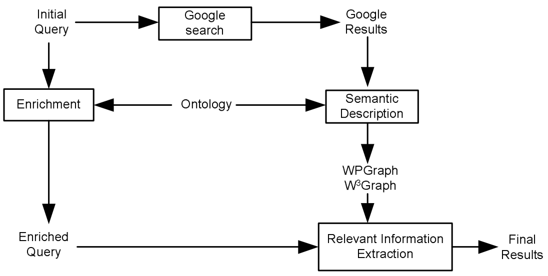

Figure1: The O^3 approach

A WPGraph is a non-oriented graph, where vertices represent concepts of a Web page and edges denote a semantic link between concepts. Edges are built according to a given ontology and are weighted using Hirst-St-Onge metrics [9]. The metrics is computed using the shortest path between two concepts and the relations between concepts along the path. Intuitively, an edge is created between two concepts if the application of the metrics does not exceed a given threshold. Similarly, a W^{3}Graph is a graph whose vertices are WPGraphs and edges are created if two WPGraphs are close from a semantic point of view. Formally, a WPGraph is a 6-tuple $(V, E, T, \varphi, \rho v, \rho e)$. Set V represents the important concepts of a Web page. Set E contains the edges; set T contains various types (video, text, image …). Function φ labels the vertices while functions ρv and ρe weight vertices and edges respectively both implementing the Hirst-St-Onge metrics. A W^{3}Graph is a triple (S, A, ρ) where set S contains vertices (here WPGraphs). Set A contains edges and function ρ, also based on the Hirst-St-Onge metrics, weights edges.

2.2. OWL primitives in the O^4 approach

The two basic OWL relations that are equivalence and subsumption are implemented in many knowledge representation languages. As regard OWL, the former relation can applied between classes and individuals while the latter only between classes (see table 1).

Instantiation is a very interesting relation for query expansion. In fact, it allows, through instances added to the initial query, to specify even more the query. Suppose, to clarify this argument, that a user wants to gain information about the weight of a Giulia (a car from the Alpha Romeo brand). Intuitively, its query will be *weight Giulia*. However, by entering such a query, most of the returned pages will contain information about persons whose name is Giulia. Therefore, the use of an ontology about the *automotive* domain can avoid this situation if the system is able to infer from the ontology that Giulia is a car. Actually, if the ontology contains *Giulia* as an instance of the *car* concept, for example, and if the latter is added to the query, the query will act as a better filter. Its interpretation will give only pages concerning the automotive domain which is in fact the user's wish. For this reason, we decided to extend, on one hand, O^3 with the instantiation relation and, on the other hand, the ASK language to offer users the possibility to specify directly in the query, the domain the query is related to.

In our approach, we also want to give users the opportunity to use adjectives in their queries. Therefore, we need to design OWL ontologies that model information indicated by adjectives. In OWL, a distinction is made between properties that link individuals to data values (Datatype property) and properties that link individuals to other individuals (Object property). If a property does not have any abstraction feature, it will not be characterized as an Object property. For example, consider the *red* colour. If there is no need to consider a variety of red (crimson, vermilion, and so on.) or red as a concept (communism for instance), red will be modelled as an attribute and will be represented using Datatype property in the ontology. Attribute values are very discriminating for a particular domain and thus are very important in our approach (e.g. there are much fewer pages that deal with red arms than pages concerning arms).

Table 1: Basic OWL relations

Relation		Abstract syntax	OWL	Logic
Equivalence	Class	EquivalentClasses (C_1,C_n)	owl:equivalentClass owl:sameAs	$C_1=\ldots=C_n$
	Instance	SameAs(I_1,I_n)	owl:sameAs owl:sameIndividual	$I_1=\ldots=I_n$
Subsumption	Class	SubClassOf(C_1,C_2)	rdfs:subClassOf	$C_1 \sqsubseteq C_2$

In table 1, OWL primitives are those of the W3C recommendation. C_i are classes and I_j are individuals.

2.3. OWL extension

Meronymy is a linguistic relation used to model an entity as a whole linked to its parts like, for example, a car (whole) related to a wheel (part). This relation is implemented in O^3 but does not exist as a basic OWL primitive. Furthermore, meronymy is a general relation which is difficult to characterize. Actually, links between object/component, member/collection or material/object can be seen as meronyms whereas the composition relation is a bit different. Therefore, we need to consider a variety of meronymy relations as in UML [13] where a distinction is made between composition and aggregation. Composition indicates that all instance of a class belongs only to one instance of another class. For example, a polygon is made of several points and if the polygon is destroyed, so are the points. Aggregation is a less rigorous way of grouping things. An order is made of several products but a product continues to exist even if the order is destroyed.

A manner to express meronymy in OWL is to use containers through RDF primitives or collection of classes using the owl:unionOf primitive. These solutions hold for classes with owl:unionOf and between individuals using containers. However, these approaches are, on one hand, definitely not intuitive and on the other hand, they increase the reasoning complexity. Moreover, it is not possible to express composition like this since individuals implemented in relations using containers or owl:unionOf can be part of several classes. Therefore, we need to define two additional OWL primitives in our approach: a first one to express composition and a second one for aggregation. The adopted semantics is inspired from the one proposed by Barbier [2]. Composition and aggregation are both transitive and asymmetric. We use cardinality constraints to differentiate them (see table 2). In the proposed axioms, C_i are OWL classes, x, y and z are individuals. For easy reading, we use the following notation: $Agg(C_1(x),C_2(y))$ instead of: $Agg(C_1,x,C_2,y)$ and $Agg(C_1,x,C_2,y) \rightarrow C_1(x) \wedge C_2(y)$ (same for composedOf).

Table 2: OWL tags for composition and aggregation

OWL Tag	Abstract syntax	First-order Logic
tg:Agg	Agg(C C’)	1) $\forall C_1, C_2, C_3\ \forall x, y, z\ Agg(C_1(x), C_2(y)) \wedge$ $Agg(C_2(y), C_3(z)) \rightarrow Agg(C_1(x), C_3(z))$ 2) $\forall C_1, C_2\ \forall x, y\ Agg(C_1(x), C_2(y))$ $\rightarrow \neg Agg(C_2(y), C_1(x))$ 3) $\forall C_1, \exists x, y, 1 \leq \lvert \{C_2(y) \mid Agg(C_1(x), C_2(y))\} \rvert$ 4) $\forall C_2 \exists x, y, 1 \leq \lvert \{C_1(x) \mid Agg(C_1(x), C_2(y))\} \rvert$
tg:composedOf	composedOf (C C’)	1) $\forall C_1, C_2, C_3\ \forall x, y, z,$ $composedOf(C_1(x), C_2(y)) \wedge$ $composed(C2(y), C3(z))$ $\rightarrow composedOf(C1(x), C3(z))$ 2) $\forall C_1, C_2\ \forall x, y, composedOf(C_1(x), C_2(y))$ $\rightarrow \neg composedOf(C_2(y), C_1(x))$ 3) $\forall C_2, \forall x, y,$ $\lvert \{C_1(x) \mid composedOf(C_1(x), C_2(y))\} \rvert = 1$ 4) $\forall C_1, \forall x, y,$ $1 \leq \lvert \{C_2(y) \mid composedOf(C_1(x), C_2(y))\} \rvert$

Opposition between classes is the other relation we have integrated to OWL. Similar to antonymy, this relation allows the specification of antagonism between concepts, relations, attributes and instances. Nevertheless, this relation is complex since it can appear under various forms. First, it can express a kind of complementarity like presence/absence or even/odd. Under this form, the affirmation of one concept of the relation implies the negation of the other one. Second, it can express antagonism between concepts that are measurable like hot/cold, small/big. This kind of opposition applies mainly between properties. Lastly, the opposition relation can affect spatio-temporal values assigned to concepts like sun/moon for instance. This last kind of opposition is interesting since it can oppose concepts like noise and silence but also relations like start and arrival. Moreover, in certain circumstances, this kind of opposition can also apply to instances like Laurel and Hardy.

Opposition is intensively used in every day's life information retrieval. However, to our knowledge, no Web search engines include this feature in their query language. They support negation but not opposition. Thus, if users are interesting to gain information about sweet cookery and if in the ontology of the cookery domain, sweet is the antonym of salty, the system should be able to understand that users are not interested in information about salty cookery. In order to define the semantics of the relation, we used axioms proposed by the linguist Edmundson [4]. He defines antonymy as being irreflexive, symmetric, antitransitive, right-identity and non-empty. As the last property would force each concept of an ontology to have an antonym we decided to leave it to one side. Table 3 hereafter gives the semantics of the OWL primitive we propose. In this definition, C_i and I_j are unary predicates, R_k are relations whereas contraryOf is a binary predicate and sameAs, equivalentClass and equivalentProperty are OWL primitives.

Table 3: Opposition in OWL

Tag	Abstract Syntax	First-order logic
tg:contraryOf	contraryOf(C_1 C_2)	1) $\forall C\ \neg \text{contraryOf}(C, C)$ 2) $\forall C_1, C_2\ \text{contraryOf}(C_1, C_2) \rightarrow \text{contraryOf}(C_2, C_1)$ 3) $\forall C_1, C_2, C_3\ \text{contraryOf}(C_1, C_2) \wedge \text{contraryOf}(C_2, C_3) \rightarrow \text{equivalentClass}(C_1, C_3)$ 4)$\forall C_1, C_2, C_3,\ \text{contraryOf}(C_1, C_2) \wedge \text{equivalentClass}(C_2, C_3) \rightarrow \text{contraryOf}(C_1, C_3)$
	contraryOf(I_1 I_2)	1) $\forall I\ \neg \text{contraryOf}(I, I)$ 2) $\forall I_1, I_2\ \text{contraryOf}(I_1, I_2) \rightarrow \text{contraryOf}(I_2, I_1)$ 3) $\forall I_1, I_2, I_3\ \text{contraryOf}(I_1, I_2) \wedge \text{contraryOf}(I_2, I_3) \rightarrow \text{sameAs}(I_1, I_3)$ 4)$\forall I_1, I_2, I_3,\ \text{contraryOf}(I_1, I_2) \wedge \text{sameAs}(I_2, I_3) \rightarrow \text{contraryOf}(I_1, I_3)$
	contraryOf(R_1 R_2)	1) $\forall R\ \neg \text{contraryOf}(R, R)$ 2) $\forall R_1, R_2\ \text{contraryOf}(R_1, R_2) \rightarrow \text{contraryOf}(R_2, R_1)$ 3) $\forall R_1, R_2, R_3\ \text{contraryOf}(R_1, R_2) \wedge \text{contraryOf}(R_2, R_3) \rightarrow \text{equivalentProperty}(R_1, R_3)$ 4)$\forall R_1, R_2, R_3,\ \text{contraryOf}(R_1, R_2) \wedge \text{equivalentProperty}(R_2, R_3) \rightarrow \text{contraryOf}(R_1, R_3)$

3. OWL ontology-based query expansion

Much work has been carried out in the area of query expansion. The main objective of query expansion or query enrichment is to add new meaningful terms to an initial query in order to improve the retrieval results. This can be done either manually, automatically or semi-automatically. In the first case, user intervention is required. It means that the system proposes a set of terms and the user chooses the most appropriate one to put in the query. For automatic query expansion, a weight function is implemented based on particular models like ontologies or corpus. This function indicates which terms are the most appropriate ones. The last approach is an hybrid form of the two others.

Concerning ontology-based query expansion, two various approaches exist in the literature. The first implements general ontology like WordNet or Cyc while the other one uses domain specific ontologies.

One of the first original query expansion approaches using WordNet is the one proposed by Voorhees [14] where the author expands the query with synonyms. In Navigli and Velardi's approach [12], more features, like gloss words or common nodes, are implemented in the expansion phase. Actually, the ontology is used to extract the semantic domain of a word, then the query is expanded using co-occurring words. Baziz et al. [1] used semantic networks and similarity measures to improve information retrieval. The major problem with WordNet-based approaches is related to the too general and vast aspect of WordNet. Terms can belong to several synsets and therefore need to be disambiguated. In O^4, since we use ontology chosen by a user that model a particular domain of interest, we don't face this problem.

Approaches using domain specific ontologies are mostly implemented in area like genomics [8], or geography [6]. In such domains, the semantic relations used to design the ontology that will serve for query expansion are very specific. In consequence, it is very relevant to make the expansion phase based on these particular relations. However, since the high specificity of such approaches, one can hardly imagine that they can apply in the context of the Web.

In our approach, we try to take advantage of both approaches. On one hand, we use an ontology representing the semantic relations of WordNet. On the other hand, since we use OWL, users are allowed either to build specific ontology or, due to the recent popularity of the language, to reuse ontologies that can be found on the Web. Therefore, the combination of OWL and the semantic relations proposed in this paper will allow users to implement adapted ontologies to target a particular domain at the right level of abstraction. Furthermore, the various approaches presented in this survey do not put the stress on any underlying query language. As far as we are concerned, we have designed the ASK query language which is tailored to query expansion and which improves even more the precision of retrieval results.

3.1. ASK query language

Queries that we propose to be enriched are written using ASK [7] whose semantics given in first-order logic is tailored for the logic structure of the WPGraphs and W^{3}Graphs. In this section, we present the adaptation of the query language with respect to the new primitives introduced in section 2. The adaptations offer users more basic operators for building queries. Furthermore, the proposed operators facilitate the query expansion process presented in the next section. Improvements concern:

1. the integration of the ":" operator that allows the user to specify in the query the targeted search domain. For instance, the query *wheel:automotive* means that the user is interested in *wheel* in the *automotive* domain. It also denotes the name of the ontology useful for query enrichment. Notice that popular search engines do not support this functionality.
2. the addition of the "-" operator to express opposition between query terms. For instance, the query *–salty:cookery* means the contrary of salty in the cookery domain.
3. the definition of specific operators to include attributes in the query. For example, the query *car.colour(=red)* targets pages concerning red cars.

3.2. Query expansion rules

The expansion rules we propose are based on the semantic relations presented above and on the properties of the logic operators of the ASK language. Work done by Joho et al [10], about the relevance of search results as regard the way initial query has been expanded, has highlighted a priority between semantic relations. Their study shows that queries expanded using synonym terms give the most satisfying results for users. Then users favour subsumption, meronymy and last opposition in this order. This is why we kept this hierarchy between relations to enrich queries. Since we also consider instances of concepts in our ontologies, we treat the instance relation at the same level than subsumption. Actually, considering our previous example, a *Giulia* can be considered as an instance of *car* but also as a kind of *car* (subsumption). The expansion is made with respect to the terms constituting the query and according to only one relation considered one after the other in the order defined above. Lastly, the name of the domain is added to the query since it can filter many irrelevant Web pages. Furthermore, logic connectors of ASK, and mainly the conjunction, are used to restrain the search domain. In fact, the query is seen as a logic formula and the added terms are constraints that must be satisfied at interpretation time. In consequence, by virtue of the precision of the returned results, users will spare much time since they will not be forced to skim irrelevant pages as it is the case with usual Web search engines. Table 4 contains the proposed rules in the order of application.

Assume, to illustrate the query enrichment process, that user wants pages related to publication about trees in the computer science domain. The following ASK query will be entered: *publication&tree:computer.* Suppose that in our ontology, *graph* and *tree* are equivalent. Therefore, according to rule 7 of table 5 the enriched query is *publication&tree&(graph|computer).* Assume now that in our ontology, no equivalent concept to graph is defined but instead a subsumption relation between *tree* and *structure*. Rule 8 will be triggered and the expanded query will be: *publication&(tree&structure)&computer.* We only use one level of abstraction in the graph representing the ontology so far. However, it is conceivable that for the future an interaction with users will be set up to target the right level of expansion for the query.

Table 4: Expansion rules

Initial query	**Enriched query**
$-\omega : O$	1) $\omega 1 \,\&\, (\omega 2 \mid \ldots \mid \omega n) \quad \forall i, 1 \le i \le n$, contraryOf$(\omega\ \omega i)$
	2) $!\omega \,\&\, O$ if there is no antonym of w in the ontology
$\omega : O$	3) $\omega \,\&\, (\omega 1 \mid \ldots \mid \omega n \mid O)$ $\forall i, 1 \le i \le n$, equivalentClass$(\omega\ \omega i) \vee$ sameAs$(\omega\ \omega i)$
	4) $\omega \,\&\, \omega 1 \,\&\, O$ if subClassOf$(\omega\ \omega 1) \vee$ InstanceOf$(\omega\ \omega i)$
	5) $\omega \,\&\, \omega 1 \,\&\, O$ if composedOf$(\omega\ \omega 1)$
	6) $\omega \,\&\, (!\omega 1) \,\&\, O$ if contraryOf$(\omega\ \omega 1)$
$\omega 1 \,\&\, \omega 2 : O$	7) $(\omega 1 \,\&\, \omega 2) \,\&\, (S1 \mid \ldots \mid Sn \mid Sn+1 \mid \ldots \mid Sm \mid O)$ $\forall i, 1 \le i \le n$, equivalentClass$(\omega 1\ Si) \vee$ sameAs$(\omega 1\ Si)$ $\forall j, n+1 \le j \le m$, equivalentClass$(\omega 2\ Sj) \vee$ sameAs$(\omega 2\ Sj)$
	8) $(((\omega 1 \,\&\, h1) \,\&\, ((\omega 2 \,\&\, h2) \mid \omega 2)) \mid (\omega 1 \,\&\, (\omega 2 \,\&\, h2))) \,\&\, O$ if (subClassOf$(\omega 1\ h1) \wedge$ subClassOf$(\omega 2\ h2)) \vee$ (InstanceOf$(\omega 1\ h1) \wedge$ InstanceOf$(\omega 2\ h2)$)
	9) $(((\omega 1 \,\&\, h1) \,\&\, ((\omega 2 \,\&\, h2) \mid \omega 2)) \mid (\omega 1 \,\&\, (\omega 2 \,\&\, h2))) \,\&\, O$ if composedOf$(\omega 1\ h1) \wedge$ composedOf$(\omega 2\ h2)$
	10) no enrichment if w_1 et w_2 are opposed $\omega 1 \,\&\, \omega 2 \,\&\, (!a1 \mid !a2)$ contraryOf$(\omega 1\ a1) \wedge$ contraryOf$(\omega 2\ a2)$

$\omega 1 \mid \omega 2 : O$	11) $(\omega 1 \mid \omega 2) \& (S1 \mid \ldots \mid Sn \mid Sn+1 \mid \ldots \mid Sm \mid O)$ $\forall i, 1 \le i \le n, \text{equivalentClass}(\omega 1\ Si) \vee \text{sameAs}(\omega 1\ Si)$ $\forall j, n+1 \le j \le m, \text{equivalentClass}(\omega 2\ Sj) \vee \text{sameAs}(\omega 2\ Sj)$
	12) $((\omega 1 \& h1) \mid (\omega 2 \& h2)) \& O$ $\text{if } (\text{subClassOf}(\omega 1\ h1) \vee \text{subClassOf}(\omega 2\ h2)) \vee$ $(\text{InstanceOf}(\omega 1\ h1) \wedge \text{InstanceOf}(\omega 2\ h2))$
	13) $((\omega 1 \& h1) \mid (\omega 2 \& h2)) \& O$ $\text{if composedOf}(\omega 1\ h1) \vee \text{composedOf}(\omega 2\ h2)$
	14) $(\omega 1 \mid \omega 2) \& O \quad \text{if contraryOf}(\omega 1\ \omega 2)$ $(\omega 1 \mid \omega 2) \& (!a1 \mid !a2) \text{contraryOf}(\omega 1\ a1) \wedge \text{contraryOf}(\omega 2\ a2)$
$\omega 1 \# \omega 2 : O$	15) $(\omega 1 \& (S1 \mid \ldots \mid Sn \mid O)) \# (\omega 2 \& (Sn+1 \mid \ldots \mid Sm \mid O))$ $\forall i, 1 \le i \le n, \text{equivalentClass}(\omega 1\ Si) \vee \text{sameAs}(\omega 1\ Si)$ $\forall j, n+1 \le j \le m, \text{equivalentClass}(\omega 2\ Sj) \vee \text{sameAs}(\omega 2\ Sj)$
	16) $(\omega 1 \& h1 \& O) \# (\omega 2 \& h2 \& O)$ $\text{if } (\text{subClassOf}(\omega 1\ h1) \wedge \text{subClassOf}(\omega 2\ h2)) \vee$ $(\text{InstanceOf}(\omega 1\ h1) \wedge \text{InstanceOf}(\omega 2\ h2)) \vee$ $(\text{composedOf}(\omega 1\ h1) \wedge \text{composedOf}(\omega 2\ h2))$
	17) $(\omega 1 \# \omega 2) \& O \quad \text{if contraryOf}(\omega 1\ \omega 2)$

4. Experimental results

In this section, we first we give a detailed illustration of the expansion process. Then, we present and discuss the results obtained with the TARGET prototype (see figure 1).

The example we give relies on the ontology depicted in figure 2. For clarity reasons, we give only the part of the ontology that is interesting for ontology expansion. It consists in submitting to Google a basic non-enriched query here: *car* obtained from the ASK query: *car:automotive*. The hundred firsts results returned by the Web search engine are transformed into WPGraphs and W^{3}Graphs. In parallel, according to table 4, the initial query is expanded and become: *car&(auto|automobile|automotive).* Lastly, the expanded query is verified on the graphs. A first observation of the final results allows us to see that all irrelevant pages like www.car.org or www.c-a-r.org and many commercial links are filtered though returned by Google. This is mainly the combined consequence of the terms added to the query and the properties of the conjunction taken into account ASK .

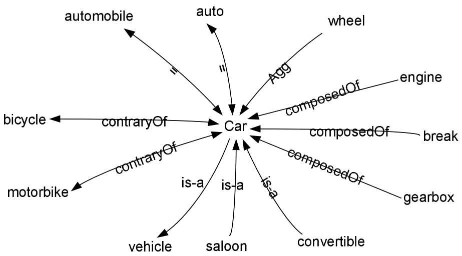

Figure 2: Subpart of the automotive ontology

In order to strengthen the result obtained with the above example and to validate all the query expansion rules, we needed to make significant experimentations. This has been done through the enrichment of the TARGET prototype (http://se2c.uni.lu/tiki/tiki-index.php?page=TargetTool) already introduced in figure 1. Our experimentation has been made based on about hundred queries. Terms added to the queries have been extracted from the ontology of the automotive domain which is partly described in figure 2. However, in order to highlight the contribution of the proposed OWL primitives, we used several variants of the same ontology so each expansion rule has been tested. Table 5 also contains comparison elements with other popular web search tools. On one hand, precision is computed with respect to the hundred first pages returned by Google and the domain modelled in the ontology. It means that if a content page matches with the domain we count the result as valid. On the other hand, recall denotes the missed pages that should have been returned.

Table 5: Experimental results

	Relations used for expansion	**Amount of returned pages**	**Precision**	**Recall**
O^4	Equivalence	47	99%	3%
	Subsumption/Instance	35	99%	10%
	Composition	26	100%	48%
	Opposition	3	100%	86%
Google		585000000	66%	
Yahoo		950000000	51%	
Alltheweb		743000000	61%	

The returned results allow us to say that O^4 is much more precise than the other popular Web search engines in terms of relevance. Moreover, among the pages proposed by usual search tools, about 40% of them concern commercial web sites. In our approach, these documents are filtered by the expanded queries and, as a result, users get only technical pages which most of the time fulfil their initial wishes. However, our approach is complex because of the WPGraphs and W^{3}Graph construction and therefore it is slower.

Expansion using the equivalence relation has shown the best results. Actually, as discussed at the beginning of this section, equivalent concepts added to the query have the effect of filtering pages that are totally out of the scope. This explains high precision and low recall.

The contribution of the subsumption and instantiation relations is a bit different. Although precision is always very high, recall is increasing. This is due to the characteristics of the relations. In fact, these implement concepts that are more specific and, as a result, pages containing only general terms denoting car are omitted. This is all the more true with terms linked with composition or aggregation relations. Since these relations are used to specify elements of a general concept, all pages dealing with general definition of the main concept as well as commercial information are filtered which explains the higher recall. This proves that such relations can be used in order to retrieve very specific information.

The ranking of the returned pages varies. Popular Web search tools use their own rank algorithm, like the well-known Google PageRank. Nevertheless, these algorithms favour commercial Web sites, this is why their pages are returned first. In the contrary, the high precision of our approach does not force users to eliminate irrelevant results.

5. Conclusion

In this paper, we have presented an extension of the OWL language as well as an illustration of an original web information retrieval application that can be made with it. The proposed approach has been formally defined and has shown a higher precision concerning the results obtained with the TARGET system than the ones obtained with popular Web search tools. Nevertheless, the Web is highly dynamic and is permanently evolving. Every day, new pages are added, other are removed or updated which make domains evolving by virtue of the diversity of the modified information. Web users are not always aware of these changes. So, further work should consider ontologies able to follow the evolution of domains with richer semantic relations and used for query enrichment. This will allow users to be even more effective when they search the Web.

6. References

[1] M. Baziz, M. Boughanem, and N. Aussenac-Gilles, “A conceptual indexing approach based on document content representation,” in *CoLIS5: Fifth International Conference on Conceptions of Libraries and Information Science – Context: nature, impact and role*, pp. 171-186, Springer-Verlag, 2005.

[2] F. Barbier, B. Henderson-Sellers, A. L. Parc-Lacayrelle, and J.-M. Bruel, “Formalization of the whole-part relationship in the unified modeling language,” *IEEE Transactions on Software Engineering*, vol. 29, pp. 459-470, May 2003.

[3] T. Berners-Lee, J. Hendler, and O. Lassila, “The semantic web,” *Scientific American*, vol. 284, pp. 34-43, May 2001.

[4] H. P. Edmundson, “axiomatic characterization of synonymy and antonymy,” in *Proceedings of the 1967 conference on Computational linguistics*, (Morristown, NJ, USA), pp. 1-11, Association for Computational Linguistics, 1967.

[5] C. D. Fellbaum, *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press, 1998.

[6] G. Fu, C. B. Jones, and A. I. Abdelmoty, “Ontology-based spatial query expansion in information retrieval,” in *ODBASE: OTM Confederated International Conferences*, November 2005.

[7] N. Guelfi and C. Pruski, “On the use of ontologies for an optimal representation and exploration of the web,” *Journal of Digital Information Management (JDIM)*, vol. 4, September 2006.

[8] W. Hersh, R. Bhupatiraju, and S. Price, “Phrases, boosting, and external knowledge resources for genomic information retrieval,” in *The Twelfth Text Retrieval Conference - TREC*, pp. 503-509, 2003.

[9] G. Hirst and D. St-Onge, “Lexical chains as representation of context for the detection and correction malapropisms,” in *WordNet: An electronic lexical database and some of its applications* (C. Fellbaum, ed.), (Cambridge, MA), pp. 305-332, The MIT Press, 1998.

[10] H. Joho, C. Coverson, M. Sanderson, and M. Beaulieu, “Hierarchical presentation of expansion terms,” in *SAC '02: Proceedings of the 2002 ACM symposium on Applied computing*, pp. 645-649, ACM Press, 2002.

[11] D. McGuinness and F. van Harmelen, “Owl web ontology language overview.” W3C Recommendation, February 2004.

[12] R. Navigli and P. Velardi, “An analysis of ontology-based query expansion strategies,” in *Proceeding of the Workshop on Adaptive Text Extraction and Mining*, (Cavtat-Dubrovnik (Croatia)), September 2003.

[13] OMG, “UML 2.0 Superstructure Specification,” Technical Report formal/05-07-04, Object Management Group, July 2005.

[14] E. M. Voorhees, “Query expansion using lexical-semantic relations,” in *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, (Dublin, Ireland), pp. 61-69, Springer-Verlag New York, Inc., 1994.