Description Logic to Model a Domain Specific Information Retrieval System

Saïd Radhouani[1], Gilles Falquet[1], and Jean-Pierre Chevalletinst[2]

[1] CUI, University of Geneva, Genève, Switzerland
{radhouani,falquet}@cui.unige.ch
[2] IPAL-CNRS, Institute for Infocomm Research, Singapore
viscjp@i2r.a-star.edu.sg

Abstract. In professional environments which are characterized by a domain (Medicine, Law, etc.), information retrieval systems must be able to process **precise queries**, mostly because of the use of a specific domain terminology, but also because the retrieved information is meant to be part of the professional task (a diagnosis, writing a law text, etc.). In this paper we address the problem of solving domain-specific precise queries. We present an information retrieval model based on description logics to represent external knowledge resources and provide expressive document indexing and querying.

1 Introduction

Information Retrieval Systems (IRS) are nowadays very popular, mainly due to the popularity of the Web. Most IRS on the Web (also called Search Engines) are general purpose, they don't take into account the specificities of the user domain of activity. We think there is a need for domain-adapted IRS: once the document domain is known, certain assumptions can be made, some specific knowledge can be used, and users may then ask much more precise queries than the usual small set of keywords in use for Web search engines.

In this work, we explore the modeling of precise search engines adapted to professional environments which are characterized by a domain: medicine, computer, law, etc. Each domain has its own terminology, i.e. its own set of terms that denote a unique concept in the domain. For example, in the medical domain, "X-ray" means an image obtained by the use of X-ray radiations, whereas in Physics domain, "X-ray" means the radiations only. In addition, users often have precise information needs that correspond to professional tasks such as writing medical reports, writing articles about specific events or situations, exploring a scientific question, etc.

In this context, the qualifier "precise" denotes a query that contains terms from a domain specific terminology and have a non trivial semantic structure. For example, a journalist would like to formulate the following query:

Query 1. Give me documents that deal with "*the US politician who won the 2007 peace Nobel prize*".

S.S. Bhowmick, J. Küng, and R. Wagner (Eds.): DEXA 2008, LNCS 5181, pp. 142–149, 2008.
© Springer-Verlag Berlin Heidelberg 2008

The journalist is looking for a *politician* whose *nationality* is *US*. A relevant document can for instance contain the name "Al Gore" without necessarily containing the terms "politician" and "US". This document may not be found by a system merely based on terms matching. A possible solution is to specify that "politician" and "US" are not the terms the user is looking for, but rather a description of the element of interest. For solving this query, the system needs some **domain knowledge** in order to infer that "Al Gore" *is a* "Politician" and that his *nationality* is "US". The underlying **query language** must also be able to allow the use of relationships for describing the user information need. Another particular case is the use of operators in the query:

Query 2. Give me "*images with a hip without any pathology*".

A relevant answer to this query must contain the *hip* and must not contain any *pathology*[1] affecting it. A relevant document may contain a *hip* without pathology together with other parts of the human anatomy affected by pathologies. For this reason, the retrieval process must ensure, that only documents containing hip affected by pathologies are excluded. This can be expressed by using a semantic relationship between the query descriptors: "hip" *affected_by* "pathology". We need domain **knowledge** during the indexing process to precisely describe the documents' content and we also need a **document language** able to allow this kind of description.

Regarding the requirements we have presented, an IR model capable to solve precise queries must involve the following interdependent components:

External resource: Solving precise queries requires domain knowledge, notably its specialised terminology and its semantic relationships.This knowledge must be expressed in a knowledge representation language and stored in an external[2] resource such as an ontology;

Expressive document language: In order to allow retrieving documents for precise queries, we need an expressive document language which allows to incorporate semantic relationships and specialised terminology within their content description;

Expressive query language: The expression of precise queries requires a query language which allows the user to explicitly use: *i*) the specialized terminology of his domain of interest; *ii*) the semantic relationships between his query descriptors and *iii*) the desired operators.

The rest of this paper is structured as follows: In Section 2, we will present the most significant approaches that use domain knowledge for information retrieval (IR). Section 3 will be dedicated to the knowledge formalism we chose for our modelling. In Section 4, we will define our IR model presenting the document model and the query model in detail. Section 5 presents our conclusions and perspectives to develop the proposed approach.

[1] No dislocation, no fracture, etc.

[2] "External" because it models knowledge which are not present in the documents (queries) to be processed, at least in an explicit and complete form.

2 External Resource Based Information Retrieval

There are mainly two categories of approaches that use ERs for IR: *conceptual indexing* [1][2][3] and *query expansion* [4][5][6]. Both of them require a disambiguation step to identify, from the ER, the concepts denoted by the words within the document and the query [7][8].

The conceptual indexing consists in representing documents (queries) by concepts instead of words [9][10][11]. Thus, during the retrieval process, the matching between a query and a document is done based on a non-ambiguous vocabulary (concepts). So far, the approaches based on this technique have not shown significant improvement in terms of retrieval performance [9][12]. One of the factors on which depends the retrieval performance is the method used to "interpret" the semantic document (query) content. In existing approaches, once the concepts have been extracted, the documents (queries) are considered as "bags of concepts". Therefore, the semantic relationships that may exist between the concepts they contain cannot be exploited. Consequently, the documents dealing with a subject close to that of the query could not be found with these approaches. Some works have shown interest in the representation of documents by semantic networks that connect the concepts of the same document. However, these networks are only used for disambiguation and not during the IR process [9]. The query expansion is a possible solution to this problem [5][6][13].

The idea behind query expansion is to use semantic relationships in order to enrich the query content by adding, from the ER, concepts that are semantically related to those of the query [5][6][13][14]. Several works analysed this aspect, but few have had positive results. In response to these failures, researchers proposed to extend the queries in a "careful" manner by selecting some specific relationships during the expansion process [4][9]. This manner allowed to improve the retrieval performance [9], but the extended queries are again considered as bags of concepts, and their structure is ignored during the retrieval process.

The existing approaches seem to be insufficient considering the requirements that we have presented. Indeed, they treat documents and queries as bags of concepts and do not sufficiently consider their structure. They are therefore incapable to solve precise queries which have complex semantic structures.

3 Formalism for Knowledge Representation

Several formalisms have been used in the IR modeling, notably Semantic Trees[15], Conceptual Graphs[16] and Description Logics (DLs)[17]. Taking into account our requirements, we found out that DLs are particularly appropriate for modeling in our context. Indeed, DLs allow to represent the three sources of knowledge (documents, queries and ER) with the same formalism, which ensures that all these sources can participate in the IR process in a uniform and effective way. This formalism provides also a high level of expressiveness, which is particularly suitable for the representation of precise information needs. Finally it offers a comparison operation that can implement the matching function of the IRS.

Description logics [18][19] form a family of knowledge representation formalisms based on logic. The basic notions of DL are atomic *concepts* and atomic *roles*. The concepts are interpreted as subsets of the individuals that constitute the domain to be modelled. The roles, are interpreted as binary relationships between individuals. Each DL is caracterised by *constructors* provided for defining complex concepts (resp. roles) from atomic concepts (roles).

Semantics. An *interpretation* I of a DL vocabulary (a set of atomic concepts and atomic roles) is a pair $(\Delta^I, .^I)$ where Δ^I is a non-empty set called the *domain of discourse* of I, and $.^I$ is a function which associates to each concept C a set $C^I \subseteq \Delta^I$, and to each role R, a binary relationship $R^I \subseteq \Delta^I \times \Delta^I$.

According to our model's requirements, we chose from existing DLs the *Attributive Language with Complements and Qualified number restrictions* ($\mathcal{ALCQ}$) language. The syntax and the semantic of the $\mathcal{ALCQ}$ language are presented in table 1. Given an atomic concept c, an atomic role R and the concept descriptions C and D, the interpretation of a complex concept is defined in table 1.

Table 1. Syntax and semantic of the $\mathcal{ALCQ}$ language

Syntax	Semantic
c	c^I
$\top$	Δ^I
$\neg C$	$\neg C^I = \Delta^I \setminus C^I$
$\bot$	$\emptyset$
$C \sqcap D$	$C^I \cap D^I$
$C \sqcup D$	$C^I \cup D^I$
$\forall R.C$	$\{d \in \Delta^I \mid \forall e \in \Delta^I (R^I(d,e) \rightarrow e \in C^I)\}$
$\exists R.C$	$\{d \in \Delta^I \mid \exists e \in \Delta^I (R^I(d,e), e \in C^I)\}$
$\geq nR.C$	$\{d \in \Delta^I \mid \lvert\{e \mid R^I(d,e), e \in C^I\}\rvert \geq n\}$
$\leq nR.C$	$\{d \in \Delta^I \mid \lvert\{e \mid R^I(d,e), e \in C^I\}\rvert \leq n\}$

A DL knowledge base is comprised of a terminological component, the *TBox*, and an assertional component, the *ABox*. The *TBox* is made of *general concept inclusion (GCI) axioms* of the form $C \equiv D$ or $C \sqsubseteq D$ where C and D are two concept expressions. For instance,

$$Parent \equiv Person \sqcap \exists hasChild\,.\,Person.$$

The ABox contains assertions of the form $C(a)$ and $R(a,b)$ where C is a concept and a and b are individual identifiers. For instance

$$Person(Jacques), Person(Maria), hasChild(Jacques, Maria)$$

Subsumption. An interpretation I satisfies the GCI $C \sqsubseteq D$ if $C^I \subseteq D^I$. I satisfiest the *TBox* T, if I satisfies all GCIs in T. In this case, I is called *model* of T. A concept D **subsumes** a concept C in T if $C \sqsubseteq D$ in every model I of T.

What makes many description logics particularly appealing is the decidability of the subsumption problem, i.e. the existence of algorithms that test if a concept subsumes another one.

4 Semantic Descriptors-Based Information Retrieval Model

We showed in Section 2 that approaches which consider documents (queries) as bags of concepts are insufficient to solve precise queries. Thus we propose to use DL expressions to represent documents and in particular the relationships that exist between the elements of a document.

4.1 The Semantic Descriptor: A New Indexing Unit

Any concept from the knowledge base may constitute a semantic descriptor when it is used withing a document (query). A semantic descriptor is an $\mathcal{ALCQ}$ expression which is intended to match as precisely as possible the concept to which it is referred to in the document (query). This expression is a conjunction of which at least one concept serves to identify the semantic descriptor. It can also contain other concepts which serve to "refine" the description of the semantic descriptor in question. Formally, a semantic descriptor S is of the form:

$$S \equiv d_{idf} \sqcap \exists described\_by.C_1 \sqcap \cdots \sqcap \exists described\_by.C_n$$

where c_{idf} is the identifying concept and $C_1, \ldots, C_n$ are the refining concepts.

The name *described_by* represents a generic relationship; in practical applications it will be replaced by a relationship (role) of the knowledge base.

Example: In a document containing "The Brazilian Minister of Sports Pelé", the semantic descriptor is identified by "Pelé" and described by "Minister of Sports" and "Brazil". Formally, this semantic descriptor is of the form:

$$S \equiv Pelé \sqcap \exists Occupation.Minister\_of\_Sports \sqcap \exists Nationality.Brazil$$

4.2 Document and Query Representation

Each document *doc* (query q) is represented by a concept R_{doc} (R_q) defined by the conjunction of the semantic descriptors belonging to *doc* (q). In order to represent the documents and the queries using semantic descriptors, we propose to use the role *indexed_by*, which allows to associate a semantic descriptor S to a given document (query) *doc* (q) to be indexed (solved). Formally, the representation R of a given document or query containing the semantic descriptors $\{S_1 \ldots S_n\}$ is an $\mathcal{ALCQ}$ expression of the form:

$$R \equiv \exists indexed\_by.S_1 \sqcap \ldots \sqcap \exists indexed\_by.S_n$$

After the indexing process, the documents index is comprised of the original *TBox* extended by the R_{doc} concepts. During the querying process, the *TBox* is extended by the concept R_q.

Examples: Query 2 (Section 1) contains two semantic descriptors (*hip*, *pathology affecting a hip*) and a negation (*without*). It is represented by:

$$R_{Q2} \equiv \exists\, indexed\_by.Hip \sqcap \neg \exists\, indexed\_by.\,(Pathology \sqcap \exists affect.Hip)$$

The query "Give me an image containing Zidane **alone**" can be represented by

$$R_{Q3} \equiv \exists\, indexed\_by.Martin\_Luther\_King \sqcap\, = 1\, indexed\_by.Person$$

Retrieval Process: The retrieval process consists in selecting the documents that satisfy the query requirements. In DL terms, the retrieval process can be seen as a task to retrieve those documents represented by concepts that are subsumed by the concept representing the corresponding query. Thus, the matching between a query q and a document *doc* is done by verifying that $R_{doc} \sqsubseteq R_q$ is true within the knowledge base. Finally, the set of relevant documents for a given query q is $\{doc \,|\, R_{doc} \sqsubseteq_T R_q\}$.

The design of the used ER has a major impact on search result. Indeed, the matching function based on the calculation of the subsumption can be very beneficial when the ER is rich in terms of *is-a* relationship. Indeed, through the algorithm that computes the subsumption, the use of DL offers a capacity of reasoning that can deduce implicit knowledge from those given explicitly in the TBox, and therefore help to retrieve relevant documents for a given query even if they do not share any words with it. However, using only the subsumption has some limits. Indeed, depending on the domain, the ER may be organized according to different semantic hierarchies. For instance, in the geographic domain, the geometric containment is probably one of the most important hierarchical relationship. The same is true for human anatomy. For example, if a user looks for a *fracture in the leg*, he or she will certainly consider a document dealing with a *pathology of the tibia* as relevant. Thus the retrieval system must take into account the *part_of* hierarchy that exists within the human anatomy. One way to solve this problem is to twist the subsumption relation and to represent the *part_of* hierarchy as a subsumption hierarchy. Thus implicitly stating, for instance, that *a tibia is a leg*. In this approach, a query

$$R_q \equiv \exists indexed\_by\,.\,(Fracture \sqcap \exists\, location\,.\,Leg)$$

will correctly retrieve a document described by

$$R_{doc} \equiv \exists indexed\_by\,.\,(Fracture \sqcap \exists\, location\,.\,Tibia)$$

because $R_{doc} \sqsubseteq R_q$ if $Tibia \sqsubseteq Leg$.

Using subsumption to mimic another relation may lead, in certain circumstances, to unexpected and conter-intuitive deductions. A "cleaner" and semantically safer approach consists in defining transitive properties to represent the

various types of hierarchies that may exist in a given domain. The above example would then lead to the following descriptors:

$$R_q \equiv \exists indexed\_by\,.\,(Fracture \sqcap \exists\, location\,.\,(\exists part\_of.Leg)$$

$$R_{doc} \equiv \exists indexed\_by\,.\,(Fracture \sqcap \exists\, location\,.\,Tibia)$$

If an axiom specifies that *part_of* is transitive and the definition of *Tibia* is of the form "... $\sqcap \exists part\_of.Leg$", then the reasoner will infer that $R_{doc} \sqsubseteq R_q$.

5 Conclusion

In order to solve precise queries, we proposed an information retrieval model based on a new indexing unit: the *semantic descriptor*. A semantic descriptor is defined by concepts and relationships, and serves to describe the semantic documents and queries content. We defined our model using the *Description Logic*, which allows a uniform precise representation of documents and queries.

In order to assess the feasibility of our approach, we conducted some experiences (not described here) on a medical document collection. The obtained results are very promising and confirmed that the use of DL has a very good impact on the retrieval performance. Indeed, the DL offers the opportunity to use background knowledge about a specific domain. Thus, during the querying process we can benefit from the powerful reasoning capabilities a reasoner offers, notably the capacity to deduce the implicit knowledge from knowledge explicitly given in the *TBox*.

It is obvious that using DL reasoners to perform IR tasks leads to performances that are several orders of magnitude slower than classical index-based IRS. Nevertheless, several issues could be worth studying to improve the DL approach performances: *i*) document descriptors are generally simple (limited to $\sqcap$ and $\exists$ constructors), thus we could devise simpler reasoning algorithms, *ii*) when queries are simple, reasoning becomes even simpler and *iii*) the document corpus is generally stable and could be pre-processed in some way to facilitate the reasoner's work.

Acknowledgments. The authors would like to thank Mathieu Vonlanthen for fruitful discussions about the use of DL reasoners to implement subsumption-based information retrieval.

References

1. Biemann, C.: Semantic indexing with typed terms using rapid annotation. In: Proceedings of the TKE 2005-Workshop on Methods and Applications of Semantic Indexing, Copenhagen (2005)
2. Mihalcea, R., Moldovan, D.: Semantic indexing using wordnet senses. In: Proceedings of the ACL-2000 workshop on Recent advances in natural language processing and information retrieval, Morristown, NJ, USA, pp. 35–45. Association for Computational Linguistics (2000)

3. Vallet, D., Fernández, M., Castells, P.: An ontology-based information retrieval model. In: GÓmez-Pérez, A., Euzenat, J. (eds.) ESWC 2005. LNCS, vol. 3532, pp. 455–470. Springer, Heidelberg (2005)
4. Qiu, Y., Frei, H.P.: Concept based query expansion. In: Korfhage, R., Rasmussen, E.M., Willett, P. (eds.) SIGIR, pp. 160–169. ACM, New York (1993)
5. Voorhees, E.M.: Query expansion using lexical-semantic relations. In: SIGIR 1994: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, pp. 61–69. Springer, Heidelberg (1994)
6. Baziz, M., Aussenac-Gilles, N., Boughanem, M.: Désambiguisation et Expansion de Requêtes dans un SRI, Etude de l'apport des liens sêmantiques. Revue des Sciences et Technologies de l'Information (RSTI) série ISI 8, 113–136 (2003)
7. Krovetz, R., Croft, W.B.: Lexical ambiguity and information retrieval. ACM Transactions on Information Systems 10, 115–141 (1992)
8. Sanderson, M.: Word Sense Disambiguation and Information Retrieval. Ph.d. thesis, University of Glasgow, Glasgow G12 8QQ, UK (1997)
9. Baziz, M.: Indexation conceptuelle guidée par ontologie pour la recherche d'information. Thèse de doctorat, Université Paul Sabatier, Toulouse, France (2005)
10. Smeaton, A., Quigley, I.: Experiments on using semantic distances between words in image caption retrieval. In: Proc. of 19th International Conference on Research and Development in Information Retrieval, Zurich, Switzerland (1996)
11. Uzuner, ö., Katz, B., Yuret, D.: Word sense disambiguation for information retrieval. In: AAAI/IAAI, p. 985 (1999)
12. Voorhees, E.M.: Natural language processing and information retrieval. In: Pazienza, M.T. (ed.) SCIE 1999. LNCS (LNAI), vol. 1714, pp. 32–48. Springer, Heidelberg (1999)
13. Mihalcea, R., Moldovan, D.I.: An iterative approach to word sense disambiguation. In: Proceedings of the Thirteenth International Florida Artificial Intelligence Research Society Conference, pp. 219–223. AAAI Press, Menlo Park (2000)
14. Baziz, M., Boughanem, M., Aussenac-Gilles, N., Chrisment, C.: Semantic cores for representing documents in ir. In: SAC 2005: Proceedings of the 2005 ACM symposium on Applied computing, pp. 1011–1017. ACM, New York (2005)
15. Berrut, C.: Une méthode d'indexation fondée sur l'analyse sémantique de documents spécialisés. Le prototype RIME et son application à un corpus médical. Thèse de doctorat, Universitè Joseph Fourier, Grenoble, France (1988)
16. Chevallet, J.P.: Un Modèle Logique de Recherche d'Informations appliqué au formalisme des Graphes Conceptuels. Le prototype ELEN et son expérimentation sur un corpus de composants logiciels. PhD thesis, Université Joseph Fourier, Grenoble (1992)
17. Meghini, C., Sebastiani, F., Straccia, U., Thanos, C.: A model of information retrieval based on a terminological logic. In: SIGIR 1993: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 298–307. ACM, New York (1993)
18. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The description logic handbook: theory, implementation, and applications. Cambridge University Press, New York (2003)
19. Brachman, R.J., Schmolze, J.G.: An overview of the kl-one knowledge representation system. In: Mylopoulos, J., Brodie, M.L. (eds.) Artificial Intelligence & Databases, pp. 207–230. Kaufmann Publishers, San Mateo (1989)