DLMedia: an Ontology Mediated Multimedia Information Retrieval System

Umberto Straccia and Giulio Visco

ISTI-CNR
Pisa, ITALY,
straccia@isti.cnr.it

Abstract. We outline DLMedia, an ontology mediated multimedia information retrieval system, which combines logic-based retrieval with multimedia feature-based similarity retrieval. An ontology layer may be used to define (in terms of a DLR-Lite like description logic) the relevant abstract concepts and relations of the application domain, while a content-based multimedia retrieval system is used for feature-based retrieval.

1 Introduction

Multimedia Information Retrieval (MIR) concerns the retrieval of those multimedia objects of a collection that are relevant to a user information need.

In this work we deal with *Logic-based Multimedia Information Retrieval* (LMIR) and follow the principles described in [9] (see [9] for an overview on LMIR literature. A recent work is also *e.g.* [6]). Let us first roughly present (parts of) the LMIR model of [9]. In doing this, we rely on Figure 1. The model has two layers addressing the multidimensional aspect of multimedia objects $o \in \mathbb{O}$ (*e.g.* objects `o1` and `o2` in Figure 1): that is, their *form* and their *semantics* (or *meaning*). The form of a multimedia object is a collective name for all its *media dependent*, typically automatically extracted features, like text index term weights (object of type text), colour distribution, shape, texture, spatial relationships (object of type image), mosaiced video-frame sequences and time relationships (object of type video). On the other hand, the semantics (or meaning) of a multimedia object is a collective name for those features that pertain to the slice of the real world being *represented*, which exists independently of the existence of a object referring to it. Unlike form, the semantics of a multimedia object is thus *media independent* (typically, constructed manually perhaps with the assistance of some automatic tool). Therefore, we have two layers, the *object form layer* and the *object semantics layer*. The former represents media dependent features of the objects, while the latter describes the semantic properties of the slice of world the objects are about. The semantic entities (*e.g.*, Snoopy, Woodstock), which objects can be about are called *semantic index terms* ($t \in \mathbb{T}$). The mapping of objects $o \in \mathbb{O}$ to semantic entities $t \in \mathbb{T}$ (*e.g.*, "object `o1` is about `Snoopy`") is called *semantic annotation*. According to the fuzzy information retrieval model (*e.g.* [2], semantic annotation can be formalized as a membership function $F\colon \mathbb{O} \times \mathbb{T} \rightarrow [0,1]$ describing the *correlation* between multimedia objects and semantic index terms. The value $F(o,t)$ indicates to which degree

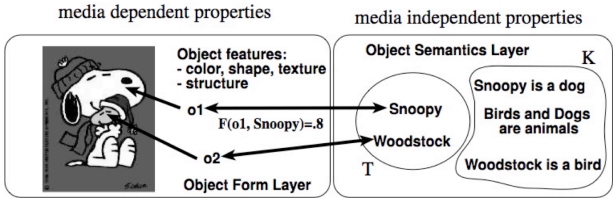

Fig. 1. LMIR model layers and objects

the multimedia object o deals with the semantic index term t. Depending on the context, the function F may be computed automatically (*e.g.*, for text we may have [4], for images we may have an automated image annotation (classification) tool, as *e.g.* [5]). Corresponding to the two dimensions of a document just introduced, there are three categories of retrieval: one for each dimension (*form-based retrieval* and *semantics-based retrieval*) and one concerning the combination of both of them. The retrieval of information based on form addresses, of course, the syntactical properties of documents. For instance, form-based retrieval methods automatically create the document representations to be used in retrieval by extracting low-level features from documents, such as the number of occurrences of words in text, or color distributions in images. To the contrary, semantics-based retrieval methods rely on a symbolic representation of the meaning of documents, that is descriptions formulated in some suitable formal language. Typically, meaning representations are constructed manually, perhaps with the assistance of some automatic tool.

A data model for MIR not only needs both dimensions to be taken into account, but also requires that each of them be tackled by means of the tools most appropriate to it, and that these two sets of tools be integrated in a principled way. Our data model is based on *logic* in the sense that retrieval can be defined in terms of logical entailment. More precisely, for computational reasons the particular logic we adopt is based on a DLR-Lite [3] like Description Logic (DL) [1]. The DL will be used in order to both define the relevant abstract concepts and relations of the application domain, as well as to describe the information need of a user. Our DL is enriched with build-in predicates allowing to address all three categories of retrieval (form-based, semantic-based and their combination). To support query answering, the DLMedia system has a DLR-Lite like reasoning component and a (feature-based) multimedia retrieval component. In the latter case, we rely on our multimedia retrieval system MILOS [1].

2 Reasoning about form and semantics in DLMedia

In order to support reasoning about form and content, DLMedia provides a logical query and representation language, which closely resembles a fuzzy variant DLR-Lite [3] with fuzzy concrete domains [10].

The concrete predicates that we allow are not only relational predicates such as $(x \leq 1500)$ (*e.g.* x is less or equal than 1500), but also similarity predicates such

[1] `http://milos.isti.cnr.it/`

as $(x\ simTxt\ 'logic, image, retrieval')$, which given a piece of text x returns the system's degree (in $[0,1]$) of being x about the keywords 'logic, image, retrieval'.

We recall that in general, a *fuzzy concrete domain* (or simply *fuzzy domain*) is a pair $\langle \Delta_{\mathtt{D}}, \Phi_{\mathtt{D}} \rangle$, where $\Delta_{\mathtt{D}}$ is an interpretation domain and $\Phi_{\mathtt{D}}$ is the set of *fuzzy domain predicates* d with a predefined arity n and an interpretation $d^{\mathtt{D}}: \Delta_{\mathtt{D}}^n \rightarrow [0,1]$. Specifically, DLMedia uses DLR-Lite($\mathtt{D}$) like axioms to describe the relevant abstract concepts of the application domain. An *axiom* is of the form $(m \geq 1)\ Rl_1 \sqcap \ldots \sqcap Rl_m \sqsubseteq Rr$ where Rl is a so-called *left-hand relation* and Rr is a *right-hand relation* with following syntax $(l \geq 1)$:

$$
\begin{array}{lll}
Rr & \longrightarrow & A \mid \exists[i_1, \ldots, i_k]R \\
Rl & \longrightarrow & A \mid \exists[i_1, \ldots, i_k]R \mid \exists[i_1, \ldots, i_k]R.(Cond_1 \sqcap \ldots \sqcap Cond_l) \\
Cond & \longrightarrow & ([i] \leq v) \mid ([i] < v) \mid ([i] \geq v) \mid ([i] > v) \mid ([i] = v) \mid ([i] \neq v) \mid \\
 & & ([i]\ simTxt\,'k_1, \ldots, k'_n) \mid ([i]\ simImg\,URN)
\end{array}
$$

where A is an atomic concept, R is an n-ary relation with $1 \leq i_1, i_2, \ldots, i_k \leq n$, $1 \leq i \leq n$ and v is a value of the concrete interpretation domain of the appropriate type. Informally, $\exists[i_1, \ldots, i_k]R$ is the projection of the relation R on the columns $i_1, \ldots, i_k$ (the order of the indexes matters). Hence, $\exists[i_1, \ldots, i_k]R$ has arity k. $\exists[i_1, \ldots, i_k]R.(Cond_1 \sqcap \ldots \sqcap Cond_l)$ further restricts the projection $\exists[i_1, \ldots, i_k]R$ according to the conditions specified in $Cond_i$. For instance, $([i] \leq v)$ specifies that the values of the i-th column have to be less or equal than the value v, $([i]\ simTxt\,'k_1 \ldots k'_n)$ evaluates the degree of being the text of the i-th column similar to the list of keywords $k_1 \ldots k_n$, while $([i]\ simImg\,URN)$ returns the system's degree of being the image identified by the i-th column similar to the object o identified by the URN (*Uniform Resource Name* [2]). We further assume that all Rl_i and Rr in $Rl_1 \sqcap \ldots \sqcap Rl_m \sqsubseteq Rr$ have the same arity. For instance assume we have a relation $Person(name, age, father\_name, mother\_name, gender)$ then the following are axioms:

$$
\begin{array}{ll}
\exists[1,2]Person \sqsubseteq \exists[1,2]hasAge & \\
 & \text{// constrains relation } hasAge(name, age) \\
\exists[3,1]Person \sqsubseteq \exists[1,2]hasChild & \\
 & \text{// constrains relation } hasChild(father\_name, child\_name) \\
\exists[4,1]Person \sqsubseteq \exists[1,2]hasChild & \\
 & \text{// constrains relation } hasChild(mother\_name, child\_name) \\
\exists[3,1]Person.(([2] \geq 18) \sqcap ([5] =' female') \sqsubseteq \exists[1,2]hasAdultDaughter & \\
 & \text{// constrains relation } hasAdultDaughter(father\_name, child\_name)
\end{array}
$$

Note that in the last axiom, we require that the age is greater or equal than 18 and the gender is female. On the other hand examples axioms involving similarity predicates are,

$$\exists[1]ImageDescr.([2]\ simImg\,urn1) \sqsubseteq Child \quad (1)$$

$$\exists[1]Title.([2]\ simTxt\,'lion') \sqsubseteq Lion \quad (2)$$

where $urn1$ is 'urn:milos:album:asantoro:image_jpeg:24d9f14a6516c95f640b47b89897b952' and identifies the image in Figure 2. The former axiom (axiom 1) assumes that we have an $ImageDescr$ relation, whose first column is the application specific image identifier and the second column contains the image URN. Then, this axiom (informally) states

[2] `http://en.wikipedia.org/wiki/Uniform_Resource_Name`

Fig. 2. The image related to $urn1$.

that an image similar to the image depicted in Figure 2 is about a $Child$ (to a system computed degree in $[0, 1]$). Similarly, in axiom (2) we assume that an image is annotated with a metadata format, *e.g.* MPEG-7, the attribute $Title$ is seen as a binary relation, whose first column is the identifier of the metadata record, and the second column contains the title (piece of text) of the annotated image. Then, this axiom (informally) states that an image whose metadata record contains an attribute $Title$ which is about 'lion' is about a $Lion$.

From a semantics point of view, given a fuzzy concrete domain $\langle \Delta_{\mathtt{D}}, \Phi_{\mathtt{D}} \rangle$, an *interpretation* $\mathcal{I} = \langle \Delta, \cdot^{\mathcal{I}} \rangle$ consists of a *fixed infinite domain* Δ, containing $\Delta_{\mathtt{D}}$, and an *interpretation function* $\cdot^{\mathcal{I}}$ that maps every atom A to a function $A^{\mathcal{I}}\colon \Delta \to [0, 1]$ and maps an n-ary predicate R to a function $R^{\mathcal{I}}\colon \Delta^n \to [0, 1]$ and constants to elements of Δ such that $a^{\mathcal{I}} \neq b^{\mathcal{I}}$ if $a \neq b$ (unique name assumption). We assume to have one object for each constant, denoting exactly that object. In other words, we have standard names, and we do not distinguish between the alphabet of constants and the objects in Δ. Furthermore, we assume that the relations have a typed signature and the interpretations have to agree on the relation's type. For instance, the second argument of the $Title$ relation (see axiom 2) is of type `String` and any interpretation function requires that the second argument of $Title^{\mathcal{I}}$ is of type `String`. To the easy of presentation, we omit the formalization of this aspect and leave it at the intuitive level. In the following, we use $\mathbf{c}$ to denote an n-tuple of constants, and $\mathbf{c}[i_1, ..., i_k]$ to denote the $i_1, \ldots, i_k$-th components of $\mathbf{c}$. For instance, $(a, b, c, d)[3, 1, 4]$ is (c, a, d). Let t be a so-called T-norm, which is a function used to combine the truth of "conjunctive" expressions. [3] Then, $\cdot^{\mathcal{I}}$ has to satisfy, for all $\mathbf{c} \in \Delta^k$ and n-ary relation R:

$$(\exists[i_1, \ldots, i_k]R)^{\mathcal{I}}(\mathbf{c}) = \sup_{\mathbf{c}' \in \Delta^n,\, \mathbf{c}'[i_1,\ldots,i_k]=\mathbf{c}} R^{\mathcal{I}}(\mathbf{c}')$$

$$(\exists[i_1, \ldots, i_k]R.(Cond_1 \sqcap \ldots \sqcap Cond_l))^{\mathcal{I}}(\mathbf{c}) = \sup_{\mathbf{c}' \in \Delta^n,\, \mathbf{c}'[i_1,\ldots,i_k]=\mathbf{c}} t(R^{\mathcal{I}}(\mathbf{c}'), {Cond_1}^{\mathcal{I}}(\mathbf{c}'), \ldots, {Cond_l}^{\mathcal{I}}(\mathbf{c}'))$$

with $([i] \leq v)^{\mathcal{I}}(\mathbf{c}') = 1$ if $\mathbf{c}'[i] \leq v$, and $([i] \leq v)^{\mathcal{I}}(\mathbf{c}') = 0$ otherwise (and similarly for the other comparison operators), while

$$([i]\, simTxt\,' k_1, \ldots, k_n')^{\mathcal{I}}(\mathbf{c}') = simTxt^{\mathtt{D}}(\mathbf{c}'[i],' k_1, \ldots, k_n') \in [0, 1]$$
$$([i]\, simImg\, URN)^{\mathcal{I}}(\mathbf{c}') = simImg^{\mathtt{D}}(\mathbf{c}'[i], URN) \in [0, 1]\,.$$

It is pretty clear that many other concrete predicates can be added as well.

Then, $\mathcal{I} \models Rl_1 \sqcap \ldots \sqcap Rl_m \sqsubseteq Rr$ iff for all $\mathbf{c} \in \Delta^n, t({Rl_1}^{\mathcal{I}}(\mathbf{c}), \ldots, {Rl_l}^{\mathcal{I}}(\mathbf{c})) \leq Rr^{\mathcal{I}}(\mathbf{c})$, where we assume that the arity of Rr and all Rl_i is n.

[3] t has to be symmetric, associative, monotone in its arguments and $t(x, 1) = x$. Examples of t-norms are: $min(x, y)$, $x \cdot y$, $\max(x + y - 1, 0)$.

Concerning queries, a *query* consists of a conjunctive query of the form $q(\mathbf{x}) \leftarrow R_1(\mathbf{z}_1) \wedge \ldots \wedge R_l(\mathbf{z}_l)$, where q is an n-ary predicate, every R_i is an n_i-ary predicate, $\mathbf{x}$ is a vector of variables, and every $\mathbf{z_i}$ is a vector of constants, or variables. We call $q(\mathbf{x})$ its *head* and $R_1(\mathbf{z}_1) \wedge \ldots, \wedge R_l(\mathbf{z}_l)$ its *body*. $R_i(\mathbf{z_i})$ may also be a concrete unary predicate of the form $(z \leq v), (z < v), (z \geq v), (z > v), (z = v), (z \neq v), (z\, simTxt'k_1, \ldots, k'_n), (z\, simImg\, URN)$, where z is a variable, v is a value of the appropriate concrete domain, k_i is a keyword and URN is an URN. Example queries are:

$q(x) \leftarrow Child(x)$
// find objects about a child (strictly speaking, find instances of $Child$)

$q(x) \leftarrow CreatorName(x, y), (y =' paolo'), Title(x, z), (z\, simTxt'tour')$
// find images made by Paolo whose title is about 'tour'

$q(x) \leftarrow ImageDescr(x, y), (y\, simImg\, urn2)$
// find images similar to a given image identified by $urn2$

From a semantics point of view, an interpretation $\mathcal{I}$ is a *model* of a rule r of form $q(\mathbf{x}) \leftarrow \phi(\mathbf{x}, \mathbf{y})$, where $\phi(\mathbf{x}, \mathbf{y})$ is $R_1(\mathbf{z}_1) \wedge \ldots \wedge R_l(\mathbf{z}_l)$, denoted $\mathcal{I} \models r$, iff for all $\mathbf{c} \in \Delta^n$:

$$q^{\mathcal{I}}(\mathbf{c}) \geq \sup_{\mathbf{c}' \in \Delta \times \cdots \times \Delta} \phi^{\mathcal{I}}(\mathbf{c}, \mathbf{c}')\,,$$

where $\phi^{\mathcal{I}}(\mathbf{c}, \mathbf{c}')$ is obtained from $\phi(\mathbf{c}, \mathbf{c}')$ by replacing every R_i by $R_i^{\mathcal{I}}$, and the T-norm t is used to combine all the truth degrees $R_i^{\mathcal{I}}(\mathbf{c}'')$ in $\phi^{\mathcal{I}}(\mathbf{c}, \mathbf{c}')$.

Finally, in DL-Media, from a conceptual point of view, we assume that the so-called set of facts is modeled as a finite set of instances of relations, *i.e.* a set of expressions of the form $\langle R(c_1, \ldots, c_n), s\rangle$, where R is an n-ary predicate, every c_i is a constant and s is the degree of truth (score) of the fact. If s is omitted, as *e.g.* in traditional databases, then the truth degree 1 is assumed. $\mathcal{I} \models \langle R(c_1, \ldots, c_n), s\rangle$ iff $R^{\mathcal{I}}(c_1, \ldots, c_n) \geq s$.

A DLMedia *multimedia base* $\mathcal{K} = \langle \mathcal{F}, \mathcal{O}\rangle$ consists of a *facts component* $\mathcal{F}$, and a *axioms component* $\mathcal{O}$. $\mathcal{I} \models \mathcal{K}$ iff $\mathcal{I}$ is a model of each component of $\mathcal{K}$. We say $\mathcal{K}$ *entails* $R(\mathbf{c})$ to degree s, denoted $\mathcal{K} \models \langle R(\mathbf{c}), s\rangle$, iff for each model $\mathcal{I}$ of $\mathcal{K}$, it is true that $R^{\mathcal{I}}(\mathbf{c}) \geq s$. The *greatest lower bound* of $R(\mathbf{c})$ relative to $\mathcal{K}$ is $glb(\mathcal{K}, R(\mathbf{c})) = \sup\{s \mid \mathcal{K} \models \langle R(\mathbf{c}), s\rangle\}$. The basic inference problem that is of interest in DLMedia is the top-k retrieval problem, formulated as follows. Given a multimedia base $\mathcal{K}$ and a query with head $q(\mathbf{x})$, retrieve k tuples $\langle \mathbf{c}, s\rangle$ that instantiate the query predicate q with maximal score, and rank them in decreasing order relative to the score s, denoted $ans_k(\mathcal{K}, q) = \mathtt{Top}_k\{\langle \mathbf{c}, s\rangle \mid s = glb(\mathcal{K}, q(\mathbf{c}))\}$.

From a reasoning point of view, the DLMedia system extends the DL-Lite/DLR-Lite reasoning method [3] to the fuzzy case (see [11]). Roughly, given a query $q(\mathbf{x}) \leftarrow R_1(\mathbf{z}_1) \wedge \ldots \wedge R_l(\mathbf{z}_l)$,

1. by considering $\mathcal{O}$ only, the user query q is *reformulated* into a set of conjunctive queries $r(q, \mathcal{O})$. Informally, the basic idea is that the reformulation procedure closely resembles a top-down resolution procedure for logic programming, where each axiom is seen as a logic programming rule. For instance, given the query $q(x) \leftarrow A(x)$ and suppose that $\mathcal{O}$ contains the axioms $B_1 \sqsubseteq A$ and $B_2 \sqsubseteq A$, then

we can reformulate the query into two queries $q(x) \leftarrow B_1(x)$ and $q(x) \leftarrow B_2(x)$, exactly as it happens for top-down resolution methods in logic programming;
2. the reformulated queries in $r(q, \mathcal{O})$ are *evaluated* over $\mathcal{F}$ only (which is solved by accessing a top-k database engine [7] and a multimedia retrieval system), producing the requested top-k answer set $ans_k(\mathcal{K}, q)$ by applying the *Disjunctive Threshold Algorithm* (DTA, see [11] for the details). For instance, for the previous query, the answers will be the top-k answers of the union of the answers produced by all three queries.

3 DLMedia at work

A prototype of the DLMedia system has been implemented. The main interface is shown in Figure 3.

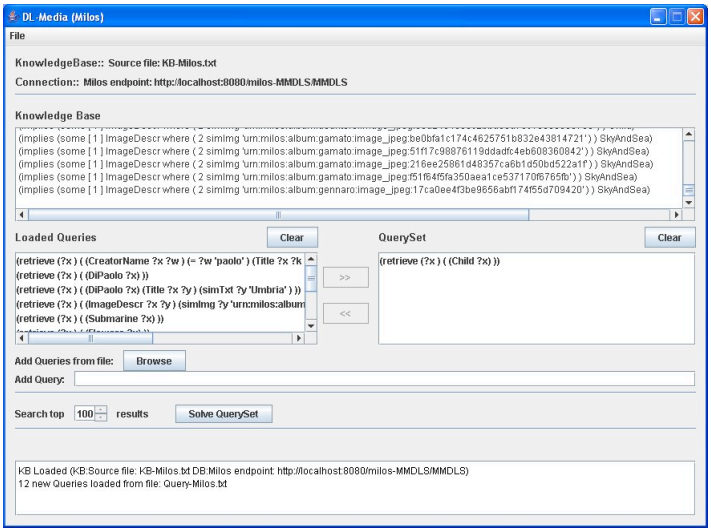

Fig. 3. DLMedia main interface.

In the upper pane, the currently loaded ontology component $\mathcal{O}$ is shown. Below it and to the right, the current query is shown ("find a child", we also do not report here the concrete syntax of the DLMedia DL).

So far, in DLMedia, given a query, it will be transformed, using the ontology, into several queries (according to the query reformulation step described above) and then the conjunctive queries are transformed into appropriate queries (this component is called wrapper) in order to be submitted to the underlying database and multimedia engine. To support the query rewriting phase, DLMedia allows also to write *schema mapping* rules, which map *e.g.* a relation name R into the concrete name of a relational table of the underlying database. The currently supported wrappers are for (of course other wrappers can be plugged in as well.)

– the relational database system Postgres; [4]
– the relational database system with text similarity MySQL; [5] and
– our multimedia retrieval system Milos, which supports XML data.

For instance, the execution of the toy query shown in Figure 3 (“find a child”) produces the ranked list of images shown in Figure 4.

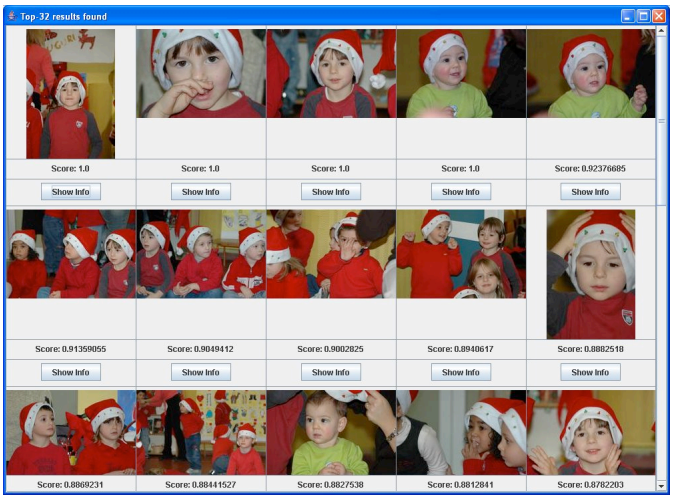

Fig. 4. DLMedia results pane.

Related to each image, we may also access to it’s metadata, which is in our case an excerpt of MPEG-7 (the data can be edited by the user as well) as shown *e.g.* in Figure 5.

Fig. 5. DLMedia image info pane.

[4] http://www.postgresql.org/
[5] http://www.postgresql.org/

4 Conclusions

In this work, we have outlined the DLMedia system, *i.e.* an ontology mediated multimedia retrieval system. Main features (so far) of DLMedia are that: (i) it uses a DLR-Lite(D) like language as query and ontology representation language; (ii) it supports queries about the form and content of multimedia data; and (iii) is scalable -though we did not address it here, query answering in DLMedia is LogSpace-complete in data complexity. The data complexity of DLMedia directly depends by the data complexity of the underlying database and multimedia retrieval engines.

References

1. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.
2. G. Bordogna, P Carrara, and G. Pasi. Query term weights as constraints in fuzzy information retrieval. *Information Processing and Management*, 27(1):15–26, 1991.
3. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Data complexity of query answering in description logics. In *Proc. of the 10th Int. Conf. on Principles of Knowledge Representation and Reasoning*, pages 260–270, 2006.
4. S. Dill, N. Eiron, D. Gibson, D. Gruhl, R. Guha, A. Jhingran, T. Kanungo, S. Rajagopalan, A. Tomkins, J.A. Tomlin, and J.Y. Zien. SemTag and Seeker: Bootstrapping the semantic web via automated semantic annotation. In *The 12th Int. World Wide Web Conference*, pages 178–186, 2003.
5. Th. Gevers and A.W.M. Smeulders. Content-based image retrieval: An overview. In *Emerging Topics in Computer Vision*. Prentice Hall, 2004.
6. S. Hammiche, S. Benbernou, and A. Vakali. A logic based approach for the multimedia data representation and retrieval. In *7th IEEE Int. Symp. on Multimedia*, pages 241–248. IEEE Computer Society, 2005.
7. C. Li, K. C. C. Chang, I. F. Ilyas, and S. Song. RankSQL: query algebra and optimization for relational top-k queries. In *Proc. of the 2005 ACM SIGMOD Int. Conf. on Management of Data*, pages 131–142, New York, NY, USA, 2005. ACM Press.
8. C. Lutz. Description logics with concrete domains—a survey. In *Advances in Modal Logics Volume 4*. King's College Publications, 2003.
9. C. Meghini, F. Sebastiani, and U. Straccia. A model of multimedia information retrieval. *Journal of the ACM*, 48(5):909–970, 2001.
10. U. Straccia. Description logics with fuzzy concrete domains. In *21st Conf. on Uncertainty in Artificial Intelligence*, pages 559–567, 2005. AUAI Press.
11. U. Straccia. Towards top-k query answering in description logics: the case of DL-Lite. In *Proc. of the 10th European Conf. on Logics in Artificial Intelligence*, pages 439–451, 2006. Springer Verlag.