A Semantic Web Multimedia Information Retrieval Engine

Miguel Bento Alves
ESTG, Instituto Politécnico de Viana do Castelo
4900-348 Viana do Castelo
Email: mba@estg.ipvc.pt

Carlos Viegas Damásio
NOVA LINCS, FCT-UNL
Universidade Nova de Lisboa
2829-516 Caparica, Portugal
Email: cd@fct.unl.pt

Nuno Correia
NOVA LINCS, FCT-UNL
Universidade Nova de Lisboa
2829-516 Caparica, Portugal
Email: nmc@fct.unl.pt

***Abstract*—We present a Semantic Web based approach that meets the requirements for multimedia information retrieval. For that, we developed a prototypical implementation using a Semantic Web framework, linking open data ontologies and image processing libraries. We show how the different mechanisms of the Semantic Web may help multimedia management, both for storage and in retrieval tasks.**

***Keywords*–Multimedia Retrieval; Semantic Multimedia; Ontologies; Semantic inference**

I. Introduction

As a consequence of technology development in several fields, large image databases have been created. In this context, well-organized databases and efficient storing and retrieval algorithms are absolutely necessary. These databases must be able to represent multimedia resources and their descriptions, considering that they are complex objects, and make them accessible for automated processing [1]. To enable multimedia content to be discovered and exploited by services, agents and applications, it needs to be described semantically. Indeed, it is very easy and cheap to take pictures, store or publish them and share, but it is very difficult and expensive to organize pictures, annotate, and find or retrieve them. This "big mismatch" between these two groups of tasks summarizes some of the key motivations for multimedia retrieval and, particularly in our work, for semantic description of visual metadata and inference on it. Multimedia constitutes an interesting field of application for Semantic Web and Semantic Web reasoning, as the access and management of multimedia content and context strongly depends on the semantic descriptions of both [1]. Semantic multimedia is a field that has emerged as a multidisciplinary topic from the convergence of Semantic Web technologies, multimedia and signal analysis. In [2], the advantages of using Semantic Web languages and technologies for the creation, storage, manipulation, interchange and processing of image metadata are described in more detail.

Manual annotation is the most effective way of doing multimedia annotation, that consists in adding metadata to the image, such as keywords or textual descriptions, to support the retrieval process. However, this method ignores the rich contents that the images have which can not be described by small sets of tags [3]. Furthermore, the keywords are very dependent on the observer [4].

A visual content feature refers to part of a visual content that contains interesting details or a property of the image which we are interested in. For any object there are many features, interesting points of the object, that can be extracted to provide a "feature" description of the object. We can have global visual content features, which describe a visual content as a whole, or local features, which represent visual content details.

In this work, our purpose is to show that a Semantic Web approach meets the requirements for multimedia information retrieval. For that, we performed an implementation using the Jena framework [5], a free and open source Java framework for building Semantic Web applications. It provides a programmatic environment for RDF, RDFS, OWL, a query engine for SPARQL and it includes a rule-based inference engine. Jena is widely used in Semantic Web applications because it offers an "all-in-one" solution for Java. Jena includes a general purpose rule-based reasoner which is used to implement both the RDFS and OWL reasoners but it is also available for general use. Jena reasoner supports rule-based inference over RDF graphs and provides forward chaining, backward chaining and a hybrid execution model. To extract visual content features we make use of Lucene Image Retrieval (LIRE) [6] [7], a light weight open source Java library for content-based image retrieval. It provides common and state-of-the-art global image features.

This document is structured as follows. We discuss the benefits of representing multimedia data in ontologies in Section II. Furthermore, we present our domain ontology. In Section III we present how to retrieve multimedia data with Semantic Web technologies and we explored the functionalities of the used Semantic Web framework to maximize the retrieval task capabilities. In Section IV we illustrate how semantic rules can be used to add knowledge to our system, reuse the knowledge produced and adding expressivity to that knowledge. We finish by discussing this work and general conclusions in Section V.

II. Storing data with ontologies

Ontologies [8], a formal representation of a set of concepts within a domain and the relationships between those concepts, are effective for representing domain concepts and relations in a form of semantic network. The motivation for the development of the MPEG-7 standard [9], an ontology for describing multimedia documents, summarizes well the use of ontologies to store multimedia data, namely, the need for a high-level representation that captures the true semantics of a multimedia object. Multimedia ontologies should provide metadata descriptors for structural and low-level aspects of multimedia documents. In [10] relevant ontologies are presented in the field of multimedia, providing a comparative study.

 ISBN: 978-1-61208-507-4

Reusing existing ontologies is always a good choice when it is appropriate to do so [11], because one of the requirements is to make sure that our system can communicate with other systems that have already committed to particular ontologies. Although it is a good practice to reuse existing ontologies, we need a domain ontology to capture the specifics of a given domain. In our work, we used the LIRE system to extract visual content features and we developed an ontology to represent the information extracted. In Figure 1, some excerpts of our LIRE ontology are presented, defining 19 global features and 5 local features. We performed mappings to the AceMedia ontology [12] and to Hunter's MPEG-7 Ontology [13], and we used Ontology for Media Resources (OMR) [14] to describe media resources. OMR is a W3C Media Annotations Working Group recommendation whose aim is to connect many media resource descriptions together, bridging the different descriptions of media resources, and provide a core set of properties that can be used to transparently express them. The mapping between our LIRE ontology and OMR allows the interoperability with other media metadata schemas, since the metadata of multimedia content can be represented in different formats. The mapping with AceMedia has the same purpose but is more focused on low-level features.

$lire{:}Feature \equiv vdo{:}Feature$
$lire{:}GlobalFeature \sqsubseteq lire{:}Feature$
$lire{:}LocalFeature \sqsubseteq lire{:}Feature$
$lire{:}EdgeHistogram \sqsubseteq lire{:}GlobalFeature$
$lire{:}EdgeHistogram \equiv mpeg7{:}EdgeHistogram$
$lire{:}ScalableColor \sqsubseteq lire{:}GlobalFeature$
$lire{:}ScalableColor \equiv mpeg7{:}ScalableColor$
$lire{:}FCTH \sqsubseteq lire{:}GlobalFeature$
$lire{:}SimpleExtractor \sqsubseteq lire{:}LocalFeature$
$lire{:}SurfExtractor \sqsubseteq lire{:}LocalFeature$
$lire{:}MPEG7features \equiv mpeg7{:}Texture \sqcup mpeg7{:}Color$
$\geq 1 lire{:}feature \sqsubseteq ma-ont{:}MediaResource$
$lire{:}globalFeature \sqsubseteq lire{:}feature$
$\top \sqsubseteq \forall lire{:}globalFeature.lire{:}Feature$
$lire{:}localFeature \sqsubseteq lire{:}feature$
$\top \sqsubseteq \forall lire{:}localFeature.lire{:}Feature$
$lire{:}edgeHistogram \sqsubseteq lire{:}globalFeature$
$\top \sqsubseteq \forall lire{:}edgeHistogram.lire{:}EdgeHistogram$
$lire{:}scalableColor \sqsubseteq lire{:}globalFeature$
$\top \sqsubseteq \forall lire{:}scalableColor.lire{:}ScalableColor$
$\geq 1 lire{:}featureProperty \sqsubseteq lire{:}Feature$
$lire{:}byteArrayRepresentation \sqsubseteq lire{:}featureProperty$
$\top \sqsubseteq \forall lire{:}byteArrayRepresentation.xsd{:}string$
$vdo{:}coefficients \sqsubseteq lire{:}featureVector$
$\top \sqsubseteq \forall lire{:}featureVector.xsd{:}string$
$lire{:}numberOfCoefficients \sqsubseteq lire{:}featureProperty$
$lire{:}numberOfBitplanesDiscarded \sqsubseteq lire{:}featureProperty$
$\geq 1 lire{:}numberOfBitplanesDiscarded \sqsubseteq lire{:}ScalableColor$
$\top \sqsubseteq \forall lire{:}numberOfBitplanesDiscarded.xsd{:}nonNegativeInteger$
$lire{:}numberOfBitplanesDiscarded \equiv vdo{:}numberOfBitPlanesDiscarded$
$lire{:}numberOfCoefficients \sqsubseteq lire{:}featureProperty$
$\geq 1 lire{:}numberOfCoefficients \sqsubseteq lire{:}ScalableColor$
$\top \sqsubseteq \forall lire{:}numberOfCoefficients.xsd{:}nonNegativeInteger$
$lire{:}numberOfCoefficients \equiv vdo{:}numberOfBitPlanesDiscarded$

Figure 1 - Excerpt of the domain ontology

The ontology can be downloaded from here [15]. Figure 2 exemplifies the descriptions of the visual features of an image. Notice that since we mapped our domain ontology to the AceMedia ontology, the same data can be obtained using the visual descriptions defined in AceMedia.

```
:im247581.jpg rdf:type ma-ont:Image ;
   lire:colorLayout [
      lire:featureVector "[34.0,20.0,28.0,...,13.0]";
      lire:byteArrayRepresentation "[21,6,34,...,13]";
      lire:numberOfCCoeff 6;
      lire:cbCoeff "[24,15,18,17,15,15,16,16,...,16]";
      lire:numberOfYCoeff 21;
      lire:yCoeff "[34,20,28,22,15,23,12,18,...,15]";
      lire:crCoeff "[40,16,11,14,17,13,...,15]" ];
   lire:scalableColor [
      lire:featureVector "[0.0,-9.0,50.0,...,-3.0]";
      lire:byteArrayRepresentation "[0,0,0,0,...,-3]";
      lire:numberOfCoefficients 64;
      lire:haarTransformedHistogram "[-129,57,...,1]";
      lire:numberOfBitplanesDiscarded 0 ];
```

Figure 2 - The descriptions of the visual features of an image

Some mappings require the use of the Jena rule engine, since they cannot be represented with OWL axioms. In Figure 3, we give an example where the property *coefficients* correspond to the property *featureVector* when this last property is defining a property of a *ScalableColor*. Notice that the inference done by the rules listed in Figure 3 cannot be modelled in OWL, namely by the OWL 2 role inclusion chain axioms, because the target is a literal. In Section IV it is explained why rules are important in the Semantic Web stack.

```
(?a lire:scalableColor ?b), (?b lire:featureVector ?c)
   -> (?b vdo:coefficients ?c).
```

Figure 3 - Rule example

III. RETRIEVING MULTIMEDIA DATA WITH SPARQL USING JENA

Sparql Protocol And Rdf Query Language(SPARQL) [16] is the language most used in Semantic Web frameworks to query RDF. SPARQL can be employed to express queries across diverse data sources, when the data is stored as RDF. The SPARQL query language is based on matching graph patterns and the results of the queries can be result sets or RDF graphs.

```
PREFIX ma-ont: <https://www.w3.org/ns/ma-ont.rdf#>
SELECT ?x ?z WHERE {
  ?x rdf:type ma-ont:Image .
  ?x lire:scalableColor ?x_sc .
  ?x_sc lire:byteArrayRepresentation ?bx .
  ex:im247581.jpg lire:scalableColor ?y_sc .
  ?y_sc lire:byteArrayRepresentation ?by .
  BIND (lire:SFDistance('ScalableColor', ?bx, ?by)
    as ?z) .
} ORDER BY ?z
```

Figure 4 - SPARQL query example with built-in functions

Copyright (c) IARIA, 2016. ISBN: 978-1-61208-507-4

The Jena framework allows the definition of SPARQL functions to be used in the query engine. A SPARQL value function is an extension point of the SPARQL query language that uses an URI to name a function in the query processor. In this way, we can develop SPARQL functions to perform operations with multimedia data. In Figure 4, we provide a SPARQL query that retrieves the distance of the *ScalableColor* feature of all images to a given image. In all code examples, both the well-know prefixes and the prefix of our ontology, *lire*, are omitted. In this example, *SFDistance* is a custom SPARQL function that calculates the distance between two images, considering the *ScalableColor* feature, represented by its byte array. This custom function developed uses the algorithm of LIRE to calculate distances between images. However, it is very easy to implement other algorithms to calculate the difference between two images and deploy them in the system as custom SPARQL functions.

Jena also provides a Java API, which can be used to create and manipulate RDF graphs. Jena has object classes to represent graphs, resources, properties and literals. In this way, we can retrieve multimedia data in Java programs without SPARQL.

IV. SEMANTIC RULES FOR MULTIMEDIA INFERENCING

It is recognised that OWL has some limitations [17]. To overcome the OWL limitations, semantic rules allow to add expressivity and expertise to our model. SWRL [18] is a proposal for representing rules/axioms for the Semantic Web, implemented by several Semantic Web frameworks. Other Semantic Web frameworks have their own rule formats, e.g., Jena framework with Jena rules [19]. Since SWRL and Jena rules are an extension of the OWL ontology language, they are restricted to unary and binary DL-predicates. In Figure 5, we give an example of how semantic rules can easily encode expert knowledge, where an image is concluded to be a member of a particular concept if some of its features are close to another image that is already classified as representing that concept.

```
(?image rdf:type ?concept) <-
  (?image lire:colorLayout ?cl1),
  (?cl1 lire:byteArrayRepresentation ?b_cl1),
  (?image lire:edgeHistogram ?eh1),
  (?eh1 lire:byteArrayRepresentation ?b_eh1),
  (?image_concept rdf:type ?concept),
  (?concept exa:minDistances ?MinD),
  (?MinD exa:minDistance ?MinD_CL),
  (?MinD_CL exa:featureClass lire:ColorLayout),
  (?MinD_CL exa:value ?min_d_cl),
  (?MinD exa:minDistance ?MinD_EH),
  (?MinD_EH exa:featureClass lire:EdgeHistogram),
  (?MinD_EH exa:value ?min_d_eh),
  (?image_concept lire:colorLayout ?cl2),
  (?cl2 lire:byteArrayRepresentation ?b_cl2),
  (?image_concept lire:edgeHistogram ?eh2),
  (?eh2 lire:byteArrayRepresentation ?b_eh2),
  ColorLayoutDistance(?b_cl1, ?b_cl2, ?Dist_cl),
  EdgeHistogramDistance(?b_eh1, ?b_eh2, ?Dist_eh),
  le(?Dist_cl, ?min_d_cl), le(?Dist_eh, ?min_d_eh).
```

Figure 5 - Semantic rule example

In the example of Figure 5, *ColorLayoutDistance* and *EdgeHistogramDistance* are custom built-in functions, provided via the Jena framework. Both use the algorithm of LIRE to calculate distances between images, as was done previously in SPARQL functions. Therefore, different algorithms can be developed and linked to the system library. In Figure 6, we give a similar example but using the work developed in [20], a system that allows the definition of SPARQL queries on Jena rules, where an image is considered an image from a given concept if there are at least 100 photos to which the features are close enough.

```
(?image rdf:type ?concept) <-
  (?image lire:colorLayout ?cl1),
  (?cl1 lire:byteArrayRepresentation ?b_cl1),
  (?image lire:edgeHistogram ?eh1),
  (?eh1 lire:byteArrayRepresentation ?b_eh1),
  (\\\SPARQL
    SELECT (COUNT(*) AS ?nCnt) WHERE {
      ?image_concept rdf:type ?concept .
      ?concept exa:minDistances ?MinD .
      ?MinD exa:minDistance ?MinD_CL .
      ?MinD_CL exa:featureClass lire:ColorLayout .
      ?MinD_CL exa:value ?min_d_cl .
      ?MinD exa:minDistance ?MinD_EH .
      ?MinD_EH exa:featureClass lire:EdgeHistogram .
      ?MinD_EH exa:value ?min_d_eh .
      ?image_concept lire:colorLayout ?cl2 .
      ?cl2 lire:byteArrayRepresentation ?b_cl2 .
      ?image_concept lire:edgeHistogram ?eh2 .
      ?eh2 lire:byteArrayRepresentation ?b_eh2 .
      BIND (lire:SFDistance('ColorLayout',
        ?b_cl1, ?b_cl2) as ?Dist_cl) .
      BIND (lire:SFDistance('EdgeHistogram',
        ?b_eh1, ?b_eh2) as ?Dist_eh) .
      FILTER (?Dist_cl <= ?min_d_cl) .
      FILTER (?Dist_eh <= ?min_d_eh) . }
  \\\SPARQL),
  ge(?nCnt, 100).
```

Figure 6 - Semantic rule with a SPARQL query

```
(?image rdf:type ?concept) <-
  (?image rdf:type ma-ont:image),
  (?beach_image rdf:type dbp_onto:Beach),
  (?image_concept rdf:type ?concept),
  (?concept exa:minDistances ?MinD),
  (?MinD exa:minDistance ?MinD_CL),
  (?MinD_CL exa:featureClass lire:ColorLayout),
  (?MinD_CL exa:value ?min_d_cl),
  (?MinD exa:minDistance ?MinD_EH),
  (?MinD_EH exa:featureClass lire:EdgeHistogram),
  (?MinD_EH exa:value ?min_d_eh),
  ColorLayoutDistance2(?image, ?beach_image,
    ?Dist_cl),
  EdgeHistogramDistance2(?image, ?beach_image,
    ?Dist_eh),
  le(?Dist_cl, ?min_d_cl),
  le(?Dist_eh, ?min_d_eh).
```

Figure 7 - Semantic rule

In Jena custom built-ins, we can use the Java API which allows to create and manipulate RDF graphs. In this way, high-level functions can be used to retrieve the data necessary to the processing inside the function. It is like a “black box”, where the details are hidden. In Figure 7, we give an example based on the previous examples but using built-in functions that receive an image as parameter instead of the lower-level data.

Finally, we present in Figure 8 a rule that infers a set of consequences starting by a set of premises and using built-in functions. In this example, the knowledge base keeps the relation of two images with respect to a given feature (*ColorLayoutDistance*).

```
(?img1 rdf:type ma-ont:Image),
(?img2 rdf:type ma-ont:Image),
ColorLayoutDistance(?img1, ?img2, ?Dist),
makeTemp(?bn) ->
   (?img1 lire:hasRelatedMediaResource ?bn),
   (?bn lire:feature lire:ColorLayout),
   (?bn lire:relatedMediaResource ?img2),
   (?bn lire:distance ?Dist).
```

Figure 8 - Semantic rule

V. Conclusions and discussion

The effectiveness of the Semantic Web technologies in the multimedia field have already been widely reported. In this work, we contribute to support the advantages of using Semantic Web languages and technologies in the multimedia field, through development of a multimedia store and retrieval system in a Semantic Web framework, namely, Jena. We gave some focus to the low-level features and we also used the LIRE system to image processing. We showed how ontologies can meet the store requirements of multimedia objects, with different mechanisms of inference that can be associated with them. It was also shown how a good design of a multimedia ontology can be useful to integrate semantic data of multimedia objects with other systems. We have shown how a powerful language as SPARQL can be useful in data retrieval. To increase the power of data retrieval, we make use of the mechanisms of the Jena framework that allow the development of multimedia custom SPARQL functions. Notice that all knowledge obtained by using the system developed in this work is open, well-known and can be shared. For example, the use of machine learning techniques to annotate multimedia content can provide a relatively powerful method for discovering complex and hidden relationships or mappings. However, it can be difficult to develop and maintain because its effectiveness depends on the design and configuration of multiple variables and options. The relationships discovered between low-level features and semantic descriptions remain hidden and are not able to be examined or manipulated [21]. The knowledge is “closed” and hidden in the systems and these systems are used as “black boxes”. We have also shown how semantic rules can increase the expertise of our system. Furthermore, it gives a better expressivity to the developer and the knowledge produced can be reused in other parts of the system or even by other systems. One of the most important purposes of multimedia systems is mapping the data produced by the visual descriptor extraction systems to higher-level semantic terms, such as objects and events. The system developed in our work is tailored to allow multimedia developers to find out these mappings. As a future work, we foresee the development of semantic rules that can represent concepts with low-level features and with a good precision and recall.

References

[1] D. J. Duke, L. Hardman, A. G. Hauptmann, D. Paulus, and S. Staab, Eds., Semantic Multimedia, Third International Conference on Semantic and Digital Media Technologies, SAMT 2008, Koblenz, Germany, December 3-5, 2008. Proceedings, ser. Lecture Notes in Computer Science, vol. 5392. Springer, 2008.

[2] R. Troncy, J. van Ossenbruggen, J. Z. Pan, and G. Stamou, “Image annotation on the semantic web,” World Wide Web Consortium, Incubator Group Report XGR-image-annotation-20070814, August 2007.

[3] R. Datta, D. Joshi, L. Li, and J. Wang, “Image retrieval: Ideas, influences, and trends of new age,” 2008.

[4] C. Ventura, “Image-based query by example using mpeg-7 visual descriptors,” Master’s thesis, 2010. [Online]. Available: http://upcommons.upc.edu/pfc/handle/2099.1/9453

[5] B. Mcbride, “Jena: A semantic web toolkit,” IEEE Internet Computing, vol. 6, no. 6, 2002, pp. 55–59.

[6] M. Lux and S. A. Chatzichristofis, “Lire: Lucene image retrieval: An extensible java cbir library,” in Proceedings of the 16th ACM International Conference on Multimedia, ser. MM ’08. New York, NY, USA: ACM, 2008, pp. 1085–1088.

[7] M. Lux and O. Marques, “Visual information retrieval using java and lire,” Synthesis Lectures on Information Concepts, Retrieval, and Services, vol. 5, no. 1, jan 2013, pp. 1–112 pp.

[8] T. Hofweber, “Logic and ontology,” in The Stanford Encyclopedia of Philosophy, spring 2013 ed., E. N. Zalta, Ed., 2013.

[9] P. Salembier and T. Sikora, Introduction to MPEG-7: Multimedia Content Description Interface. John Wiley & Sons, Inc., 2002.

[10] M. C. Suárez-Figueroa, G. A. Atemezing, and O. Corcho, “The landscape of multimedia ontologies in the last decade,” Multimedia Tools Appl., vol. 62, no. 2, Jan. 2013, pp. 377–399.

[11] L. Yu, A Developer’s Guide to the Semantic Web. Springer, 2011.

[12] S. Bloehdorn, K. Petridis, C. Saathoff, N. Simou, Y. Avrithis, S. H, Y. Kompatsiaris, and M. G. Strintzis, “Semantic annotation of images and videos for multimedia analysis,” In Proc. of the 2nd European Semantic Web Conference, ESWC 2005, 2005, pp. 592–607.

[13] J. Hunter, “Adding Multimedia to the Semantic Web - Building an MPEG-7 Ontology,” in Proceedings of the 1st International Semantic Web Working Symposium, Stanford, USA, August 2001.

[14] P.-A. Champin, T. Bürger, T. Michel, J. Strassner, W. Lee, W. Bailer, J. Söderberg, F. Stegmaier, J.-P. EVAIN, V. Malaisé, and F. Sasaki, “Ontology for media resources 1.0,” W3C, W3C Recommendation, Feb. 2012, http://www.w3.org/TR/2012/REC-mediaont-10-20120209/.

[15] https://github.com/mbentoalves/LIRE_Ontology/blob/master/lire.ttl

[16] E. Prud’hommeaux and A. Seaborne, “SPARQL Query Language for RDF,” W3C, Tech. Rep., 2006. [Online]. Available: http://www.w3.org/TR/rdf-sparql-query/

[17] B. Parsia, E. Sirin, B. C. Grau, E. Ruckhaus, and D. Hewlett. Cautiously approaching swrl. Preprint submitted to Elsevier Science. [Online]. Available: http://www.mindswap.org/papers/CautiousSWRL.pdf (2005)

[18] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean, “SWRL: A semantic web rule language combining OWL and RuleML,” World Wide Web Consortium, W3C Member Submission, 2004. [Online]. Available: http://www.w3.org/Submission/SWRL

[19] Jena Documentation. Reasoners and rule engines: Jena inference support. [Online]. Available: http://jena.apache.org/documentation/inference/

[20] M. B. Alves, C. V. Damásio, and N. Correia, “SPARQL commands in jena rules,” in Knowledge Engineering and Semantic Web - 6th International Conference, KESW 2015, Moscow, Russia, September 30 - October 2, 2015

[21] L. Hollink, S. Little, and J. Hunter, “Evaluating the application of semantic inferencing rules to image annotation,” in Proceedings of the 3rd international conference on Knowledge capture, ser. K-CAP ’05. New York, NY, USA: ACM, 2005, pp. 91–98.