OntoDoc: an ontology-based query system for digital libraries

Luigi Cinque, Alessio Malizia, Roberto Navigli
Computer Science Dept., University ”La Sapienza” of Rome
(cinque,malizia,navigli)@di.uniroma1.it

Abstract

The decreasing cost and the increasing availability of new technologies is enabling people to create their own digital libraries. One of the main topic in personal digital libraries is allowing people to select interesting information among all the different digital formats available today (pdf, html, tiff, etc.). Moreover the increasing availability of these on-line libraries, as well as the advent of the so called Semantic Web [1], is raising the demand for converting paper documents into digital, possibly semantically annotated, documents. These motivations drove us to design a new system which could enable the user to interact and query documents independently from the digital formats in which they are represented. In order to achieve this independence from the format we consider all the digital documents contained in a digital library as images. Our system tries to automatically detect the layout of the digital documents and recognize the geometric regions of interest. All the extracted information is then encoded with respect to a reference ontology, so that the user can query his digital library by typing free text or browsing the ontology.

1. Introduction

The main goal of our system, OntoDoc, is to allow users to query their own personal digital libraries in an ontology-based fashion. An ontology [9] specifies a shared understanding of a domain of interest. It contains a set of concepts, together with its definitions and interrelationships, and possibly encodes a logical layer for inference and reasoning. Ontologies play a major role in the context of the so called *Semantic Web* [1], Tim Berners-Lee's vision of the next-generation Web, by enabling semantic awareness for online content.
OntoDoc uses a reference ontology to represent a conceptual model of the digital library domain, distinguishing between text, image and graph regions of a document, providing attribute relations for them, like size, orientation, color, etc.

In order to classify a document, OntoDoc performs a first layout analysis phase, generating a structured, conceptual model from a generic document. Then the conceptual model goes through an indexing phase based on the features in the model itself. Finally, the user can query his digital library by typing free text or through composition of semantic expressions. The query system is particularly suited to express perceptual aspects of intermediate/high-level features of visual content, because the user does not have to bother thinking in terms of inches, RGB components, pixels, etc. Instead, the user can query the system with higher level, although accurate, concepts (e.g. medium size, black color, horizontal orientation etc.).
In Section 2 we explain the layout segmentation phase. In Section 3 the reference ontology is detailed. In section 4 the query system, as well as an example, is presented. In the final section we discuss conclusions and future works.

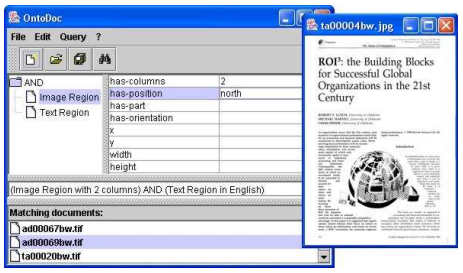

Figure 1. The System window, with a matched digital document on the right.

2. Layout analysis architecture overview

Our system [2] uses a split and merge technique similar to the approach that has been obtained by Nagy's X-Y cut algorithm, but instead of working top-down, we use the recognized horizontal and vertical lines to cut the image into small regions, we then try to merge from bigger regions, using a quad-tree technique and image processing algorithms.

The system is based on these different phases in order to perform document classification in a modular and efficient way. The system architecture includes four main components: the preprocessor (1), the split module (2), the merge module (3) and the classification module (4). All these modules can be grouped in a Layout Analyzer module which takes a document image as input and outputs a structured, conceptual description of the regions contained in the document. Actually the "brain" of our system is the classification submodule of the Layout Analyzer module, which outputs the structured model of the digital document, i.e. segmented regions together with their attributes encoded in an ontological format.

2.1. Preprocessing

The pre-processing phase performs two steps: it loads the scanned image in main memory and computes the gray-level histogram extracting the three parameters discussed below. This approach is due to the fact that we want to reduce the amount of computations in the preprocessing phase obtaining a quick response method. Starting from a 256 gray level document we compute the RI1 parameter, as the maximum value in the first half of the gray-level histogram. Then for the RI2 parameter we compute it in the same way but on the other half of the histogram, thus considering the whitening colors. Finally, the last parameter RI3, which represents the point of separation between background and text, is the minimum between the RI1 and RI2 parameter.

2.2. Split

During the split phase, the whole image of the document is split according to the extracted vertical and horizontal lines as well as the boundaries of recognized images [7]. This results in many small zones (block sizes are within a range depending on the size of the image). We use a quadtree decomposition to perform spatial segmentation by assigning a condition by which nodes are split, which is based on the average block sizes. In order to choose whether to split or not a region we use the mean and variance values for that region. If the variance is low compared to the entire document, probably the region is an image, because both characters and graphs have a high variance since they usually don't have smooth colors (i.e. the foreground color is very different from the background). After this step, labels are assigned to regions in order to pre-classify them. This pre-classification is useful to pass information to the merge phase. We define three classes of regions: *text-graph*, *image*, *background*. In case of low variance and low mean values we label the region as *background*, instead if we have high variance and low mean we label the region as *text-graph*; otherwise the label is *image*.

2.3. Merge

The split operation results in a heavily over-segmented image. The goal of the merge operation is to join neighboring zones to form bigger rectangular zones. The first phase of merging consists of connecting neighbor regions with the same pre-classification value. Using only pre-classification we don't have all the information we need, but with this approach we follow one of the targets of our method, the computing performance efficiency. The second step of the merging phase is the Union phase. The Union procedure is used to enhance the pre-classification results. First of all, the regions, which are in to the external edges of the document, are removed, then all the other regions are considered for the further phase. We now group all the Macro-Regions, as those regions with a spanned area greater than a threshold, which is based on the average region sizes. All the adjacent Macro-Regions with the same pre-classification values are merged thus obtaining our segmentation. Then we introduce the $0 \leq P(C_i|M) \leq 1$ as the estimation of the conditional probability for the given Macro-Region M of belonging to the class C_i, where $|C| = 3$ and $C = \{Text, Graph, Image\}$. Let $|M| = m$ be the total number of subregions of M, and m_T the number of subregions of M with pre-classification $\{text - graph\}$ and variance highest than the average variance overall the sub-regions of M with pre-classification $\{text - graph\}$; let m_G be the number of subregions of M with pre-classification $\{text - graph\}$ and variance lowest than the average variance overall the sub-regions of M with pre-classification $\{text - graph\}$. Let m_I be the number of subregions of M with pre-classification $\{image\}$, then: $P(C_0|M) = \frac{m_T}{m}$, $P(C_1|M) = \frac{m_G}{m}$, $P(C_2|M) = \frac{m_I}{m}$, which are respectively the probability of a Macro-Region M of belonging to the class: text, graph or image. The Macro-Regions are labeled as belonging to the class according to the highest probability as defined above. After that, the system produces an OWL (Ontology Web Language[1]) description of the Macro-Regions, which maps the digital document in input to our conceptual model for digital libraries, i.e. the reference ontology illustrated in the next subsection.

The structured model obtained after the classification will contain different instances of region elements depending on the classification results. This information will be formatted in OWL according to the template given below.

Example of an OWL file produced by our system

1 http://www.w3.org/TR/owl-features/

Figure 2. A portion of the reference ontology for digital libraries.

```
<?xml version="1.0"?>
  <!-- OWL snippet -->
  <Text_Region rdf:ID="text_region1">
    <font-size>
      <Medium_Size rdf:ID="medium_size1" />
    </font-size>
    <has-orientation>
      <Horizontal_Orientation
       rdf:ID="horiz_orient1"/>
    </has-orientation>
    <text-color>
      <Black rdf:ID="black1" />
    </text-color>
  </Text_Region>
```

3. The Reference Ontology

The reference ontology encodes all the required information about the digital library domain. Documents and Regions are represented as resources with a number of attributes in common, e.g. Orientation, Size, etc. The ontology encodes specific relations between subconcepts of Region, i.e. Text, Image and Graph Regions, and several attribute concepts, like Color, Size, Orientation, Reading Direction and so on (a snapshot of the concept taxonomy is shown in Figure 2).
Encoding ontological concepts instead of numerical attributes is a peculiar feature of our system. The user can think of concepts instead of low level measures (e.g. inches, pixels etc.) and submit queries like ”all the documents with a text region in the south part and an image region with 2 columns having an inner table”. The query is then translated in OWL and matched against the document base. The user benefits from the use of a reference ontology in that the annotated document base can be put online and queried by other users referring to the same or another ontology if a mapping is provided. In order to facilitate this task, we mapped each concept of our ontology to WordNet [3], a de facto standard lexicalized ontology containing more than 120,000 concepts. WordNet encodes each concept as a set of English synonyms, called *synset*. This allows our system to accept free text queries and automatically map each keyword to a concept in the reference ontology. For the Italian language we used MultiWordNet [8], an Italian version of WordNet.

4. Querying a personal digital library with OntoDoc

OntoDoc allows users to query their own digital libraries either by composing semantic expressions through ontology browsing or by typing some text in natural language.
The first option allows the user to browse the ontology and instantiate concepts of kind Resource, i.e. Document, Text Region, Graph Region and Image Region. For each instance (left frame in the left window of Figure 1) the user can fill the relation range slot with attribute instances, namely size, colors, reading direction, language, orientation etc. (right frame in the left window of Figure 1). Notice that these are not quantities, but instances of concepts, so for instance the user does not choose the number of points for a font size, but he/she instantiates the Medium Size concept.
The second option is an interface for building semantic queries through keyword typing. The interpreter either maps each keyword to a lexical item in a WordNet synset or marks it as unknown. In the first case, the interpreter translates the WordNet synset to the associated concept in the reference ontology, otherwise it discards the keyword (see Figure 3 for an example). The user can also connect keywords with logic connectives like *or*, *and*, etc. Finally, the system instantiates each concept and fills in the gaps, formulating a semantic query.

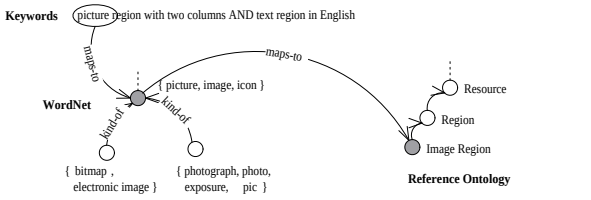

Figure 3. The mapping steps from a keyword to a WordNet synset to a reference ontology concept.

For instance, consider the query shown in Figure 3: ”picture region with two columns AND text region in English”. The interpreter assigns a synset in WordNet to each meaningful word, obtaining the following string: ”picture#1 region#1 with two#1 columns#3 AND text#1 region#1 in English#1” (the number of the correct sense in WordNet

is attached to each word). Then, the interpreter translates each synset to a reference ontology concept, obtaining: "*Image Region* with 2 *has-columns* AND *Text Region* in *English*" (concepts and relations are marked in italic). Finally, concepts are instantiated and relations are associated with these instances. The resulting query is: image-region#1, text-region#1, has-columns(image-region#1, 2) and language(text-region#1, English). Notice that the result of either query composition by ontology browsing or natural language input is a semantic expression encoded in OWL and used by the system to query the document base and return the matching documents.

5. Experimental results and Conclusions

We have tested our system over the UW-II database that is the second in series of document image databases produced by the Intelligent Systems Laboratory, at the University of Washington, Seattle, Washington, USA. This database was particularly useful because it contains 624 English journal document pages (43 complete articles) and 63 MEMO pages. All pages are scanned pages.

Each document in the database has been taken from scientific journals and contains text, graphs and images. All the images were already annotated with labels for the region type (image, text, ...) and sizes. The experiments have been carried out in 2 phases: the first phase was to the test our layout analysis module over the UW database to verify the percentage of the automatic classification of the digital documents regions; while the second phase was performed on 10 users in order to measure the ability of the system in helping them to retrieve documents.

For the first experiment, concerning the classification abilities of our system, we have tested it over the entire database (600 images) obtaining an 84% of correctly recognized regions, 14% of incorrectly recognized and 2% to be defined. The 84% of correctly recognized regions could be subdivided into a 59% of entirely recognized and a 25% of partially recognized, which means that some regions were assigned to the right class and some others not, for example a single text region was interpreted as two text regions (this usually happens in titles with many spaces).

For the second phase, all tests have been carried out using the relevance feedback process by which the user analyzes the responses of the system and indicates, for each item retrieved, a degree of relevance/non-relevance or the exactness of the ranking [5]. Annotated results are then fed back into the system, to refine the query so that new results are more fitting. The experiment was implemented showing to the users 10 different documents and then asking them to retrieve the documents from the entire UW database using our query module. Our system proved to be highly effective because the users concentrated on the conceptual content of documents rather than on numerical information about them, allowing faster and more accurate retrieval of the desired documents with respect to keyword-based non-ontological retrieval.

A major improvement of OntoDoc may be in the classification phase. In fact, the system could classify shapes like subject images or specific geometry on the basis of their ontological descriptions (for instance, finding a document with an image of an apple, or with a pie-chart).

Furthermore, mapping the reference ontology to WordNet could allow to make inferences like: *bitmap* $\overset{kind-of}{\rightarrow}$ *image* and accept *bitmap region* as input instead of *image region*.

In the next version, we plan to include an inference system based on the rules described in [6]. This will improve the expressiveness of the natural language interpreter described in Section 4.

Finally, we plan to use OntoLearn [4], a tool for ontology learning, to enrich the reference ontology with new concepts and relations extracted from a corpus of documents like the ones used for the ICDAR 2003 ([2]) page layout competition.

References

[1] T. Berners-Lee. *Weaving the Web*. Harper, SF, 1999.

[2] L. Cinque, S. Levialdi, and A. Malizia. An integrated system for the automatic segmentation and classification of documents. In ACTAPress, editor, *Proceedings of the International Conference on Signal Processing, Pattern Recognition, and Applications (SPPRA 2002)*, pages 491–496, 2002.

[3] A. Miller. Wordnet: An on-line lexical resource. *Journal of Lexicography*, 4(3), 1990.

[4] M. Missikoff, R. Navigli, and P. Velardi. An integrated approach for web ontology learning and engineering. *IEEE Computer*, pages 60–63, November 2002.

[5] G. Nagy. Twenty years of document image analysis. *PAMI, IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(1), 2000.

[6] R. Navigli and P. Velardi. Semantic interpretation of terminological strings. In F. INIST-CNRS, andoeuvre-ls-Nancy, editor, *Proc. 6th Int'l Conf. on Terminology and Knowledge Engineering (TKE 2002)*, pages 95–100, 2002.

[7] T. Pavlidis. *Algorithms for Graphics and Image Processing*. Computer Science Press, Rockeville, MD, 1982.

[8] E. Pianta, L. Bentivogli, and C. Girardi. Multiwordnet: developing an aligned multilingual database. In I. Mysore, editor, *Proceedings of the First International Conference on Global WordNet*, January 2002.

[9] B. Smith and C. Welty. Ontology: towards a new synthesis. In A. Press, editor, *Proc. of Formal Ontology in Information Systems FOIS-2001*, 2001.

2 http://www.essex.ac.uk/ese/icdar2003/