Automatically Joining Pictures to Multiple Taxonomies*
(Extended Abstract)

Ilaria Bartolini and Paolo Ciaccia

DEIS, Università di Bologna - Italy
{i.bartolini,paolo.ciaccia}@unibo.it

Abstract. Automatically providing semantics to multimedia objects is still a major open problem. In this paper we describe recent advances within this context and how they have been implemented within the Scenique image retrieval and browsing system. Scenique is based on a multi-dimensional model, where each dimension is a tree-structured taxonomy of concepts, also called semantic tags, that are used to describe the content of images. We describe an original algorithm that, by exploiting low-level visual features, tags, and metadata associated to an image, is able to predict a high-quality set of semantic tags for that image.

1 Introduction

Automatic image annotation aims to enable text-based techniques (search, browsing, clustering, classification, etc.) to be applied also to objects that otherwise could only be dealt with by relying on feature-based similarity assessment, which is known to be inherently imprecise [11]. Approaches to automatic image annotation include a variety of techniques, and they even differ in what "annotation" actually means, ranging from enriching images with a set of keywords (or *tags*) [8, 1, 6, 7], to providing a rich semantic description of image content through the concepts of a full-fledged RDF ontology [10]. Further, solutions may differ in what kind of tags/concepts they ultimately provide, in this case the difference being among general-purpose systems and others that are tailored to discover only specific concepts/classes [9, 12].

In this paper we present Ostia (Optimal Semantic Tags for Image Annotation), a novel image annotation method that predicts for an image a set of so-called *semantic tags*, i.e., concepts taken from a set of tree-structured taxonomies (also called *classification hierarchies*). Semantic tags can be regarded as a means to describe images that is more precise and powerful than free tags (with no inherent semantics), yet not so complex to derive as concepts of RDF-like ontologies (whose semantics might not be so easy to grasp by end-users). Figure 1 provides an intuition on the problem we deal with: Given an image, possibly coming with some textual description, and a set of taxonomies, the objective is to predict which are the concepts in such taxonomies that better describe the image. We have implemented Ostia within our Scenique searching

* This work is partially supported by the CoOPERARE MIUR Project.

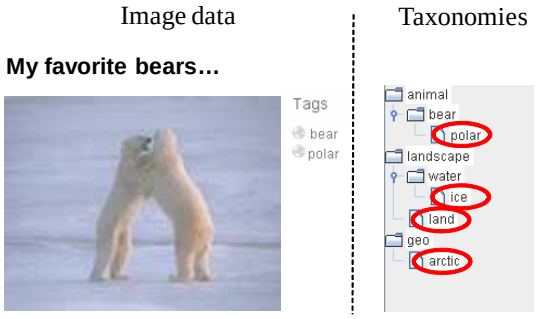


Fig. 1. For the image on the left, predicted semantic tags (on the right) are `animal/bear/polar`, `landscape/water/ice`, `landscape/land/`, and `geo/artic`.

and browsing system and tested over a real-world collection of 100,000 images. The preliminary results we report in Section 4 demonstrate that our approach can be highly effective in predictive relevant semantic tags.

2 The Problem

Scenique [2] is an integrated searching and browsing system that allows images to be organized and searched along a set of orthogonal *dimensions* (also called *facets*). Each dimension is organized as a tree and can be viewed as a particular coordinate used to describe the content of images. Scenique supports both *semantic* and *visual* facets, the latter being used to organize images according to their low-level features and not relevant in this paper.

A semantic dimension D_h, $h = 1, \ldots, M$, is a tree-structured taxonomy of concepts, also called *semantic tags*. More precisely, a semantic tag st_j is a path in D_h, $st_j = n_0/n_1/\ldots/n_k \in D_h$, where each n_i is a node of the taxonomy. Node n_i has a label that, for the sake of simplicity, we also denote as n_i.[1] The label of the root node is the dimension name (e.g., `location`, `subject`, etc.)

In the scenario we consider, Scenique manages an image database $\mathcal{DB} = \{I_1, \ldots, I_N\}$ and a set of M dimensions $D_1, \ldots, D_M$. In the more general case, an image $I_i \in \mathcal{DB}$ has the following components: A source file P_i (e.g., a JPEG picture); a set of low-level visual *features* F_i automatically extracted from P_i; the image *metadata* M_i, of which for the purpose of this paper we consider only the title, a textual description and a set of free tags (some, or even all, of these metadata might be missing for an image); a set of *keywords* $K_i = \{kwd_{i,j}\}$, automatically derived from M_i; and a set of *semantic tags* $ST_i = \{st_{i,j}\}$. Thus, each image I_i can be concisely represented as $I_i = (P_i, F_i, M_i, K_i, ST_i)$. The problem we consider can be concisley stated as follows:

[1] This is only to simplify the presentation: Scenique allows the same label to be attached to multiple nodes, e.g., `activity/sport/soccer/Italy` and `activity/sport/basket/Italy`.

Problem 1 *Given an image database $\mathcal{DB}$ and a (query) image $Q = (P, M)$ (i.e., $F = K = ST = \emptyset$), determine the set of m ($m \geq 1$)* semantic tags *ST that better describe the content of the image Q.*

3 The Ostia Algorithm

We adopt a 2-step approach to solve Problem 1, as illustrated in Figure 2 (a). First, a set of low-level visual features F and high-quality keywords K are extracted from Q. To this end we use, respectively, the feature extraction algorithm of the Windsurf library [3], which characterizes an image with color and texture features, and text analysis procedures, such as stemming, stoplist, and NLP [4] techniques,[2] not further described here for lack of space.

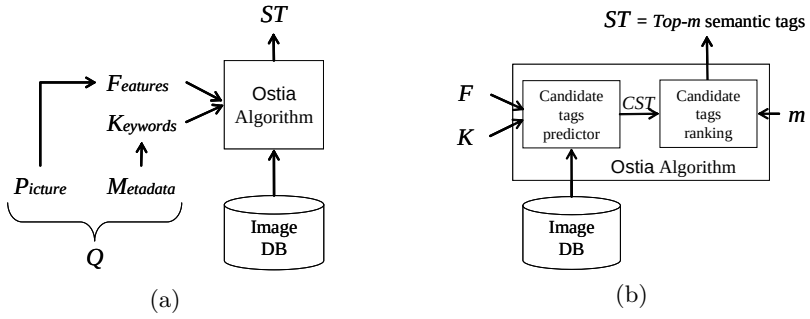


Fig. 2. Illustration of the approach (a) and of the modules of the Ostia algorithm (b).

Once both F and K have been extracted, they are input to an algorithm, called Ostia, that exploits information associated to images in the $\mathcal{DB}$ that are *similar* to Q either at the visual or the textual level (or both), to predict a set of semantic tags for Q. Ostia consists of two main modules, see Figure 2 (b). A first module is in charge of predicting a superset of ST, which are hereafter called *candidate semantic tags* (or simply candidates) and denoted CST. A second module organizes, for each dimension D_h, the candidates into a *candidate tree* $CT_h \subseteq D_h$, ranks them, and returns the Top-m ones.

3.1 Generating Candidate Semantic Tags

The first module of Ostia predicts, for each dimension D_h, a set of candidate semantic tags CST_h, with $CST = \bigcup_{h=1}^{M} CST_h$. The basic rationale of CST_h computation is to exploit available information of the query Q (i.e., K and F) in order to find images $I_i \in \mathcal{DB}$ that might contain tags relevant for Q.

We exploit query keywords K by applying a *co-occurrence* search on $\mathcal{DB}$ image keywords. The search provides a set of images that share at least e terms with Q. We rank the images on the base of the co-occurrence value and, for the top-p images only, their keywords are added to a set RK of *relevant keywords*

[2] OpenNLP: `http://opennlp.sourceforge.net/`

(which by default includes all keywords in K), and all the semantic tags are used to initialize CST. For example, if $K = \{\texttt{beach}, \texttt{sea}\}$, $e = 2$, and there is an image I_i with $K_i = \{\texttt{beach}, \texttt{sea}, \texttt{sky}\}$ and $ST_i = \{\texttt{landscape/water/sea}\}$, then `sky` is added to RK and `landscape/water/sea` to CST.

Starting from the query features F, a *nearest-neighbors* search is performed on the $\mathcal{DB}$, which determines the set of the g images most similar to Q. For all keywords kwd_j (resp. semantic tags st_j) associated to at least one of such images, a frequency score is computed as the number of top-g images annotated with kwd_j (resp. st_j). Such annotations are then ranked based on their frequency and the top-s ones are added to RK and CST, respectively.

After the above-described steps, each relevant keyword $kwd_j \in RK$ is processed, since it can provide new candidate semantic tags. For each kwd_j we check if there is any path (i.e., semantic tag) st_j in the taxonomy of some dimension D_h terminating with a label equal to kwd_j (we call this step *joining phase*). If this is the case, st_j is added to CST_h and kwd_j deleted from RK.

For keywords that, after the joining phase, still populate RK, we apply a *keyword expansion* step in order to verify if it is possible to collect further semantic tags by means of *correlated* terms (namely, synonyms) available from WordNet.[3] For instance, if $\texttt{sea} \in RK$ and the label `sea` is not part of any dimension, whereas the semantic tag $st_j = \texttt{landscape/water/ocean}$ appears in some D_h, then st_j will be added to CST_h. For each keyword $kwd_j \in RK$, we find the matching lexical concept in WordNet, collect the synonyms of the associated synsets, add them to RK, and then apply to them the joining phase.

Algorithm 1 summarizes the above steps. Notice that, since in general a semantic tag st_j can be predicted multiple times, we keep trace of its *frequency*, $freq_j$, which will be used by the second module of Ostia.

Algorithm 1 Ostia: Candidate Semantic Tags Predictor

Input: $Q = (F, K)$: query image, $\mathcal{DB}$: image database, e, p, g, s: integer
Output: CST: candidate semantic tags

$CST \leftarrow \emptyset$, $RK \leftarrow K$;
COImg $\leftarrow$ Top(KwdSearch($K, \mathcal{DB}, e$), p); ▷ *Top-p images sharing $\geq e$ kwd's with Q*
$RK \leftarrow RK \cup \{kwd_{i,j} : I_i \in \text{COImg}\}$;
$CST \leftarrow CST \cup \{st_{i,j} : I_i \in \text{COImg}\}$;
NNImg $\leftarrow$ NNImgSearch($F, \mathcal{DB}, g$); ▷ *Top-g most similar images to Q*
$RK \leftarrow RK \cup$ Top($\{kwd_{i,j} : I_i \in \text{NNImg}\}, s$); ▷ *Top-s freq.-based keywords*
$CST \leftarrow CST \cup$ Top($\{st_{i,j} : I_i \in \text{NNImg}\}, s$); ▷ *Top-s freq.-based semantic tags*
$CST \leftarrow CST \cup$ Joining($RK, \{D_h\}$); ▷ *join keywords in RK to paths in some D_h*
$RK \leftarrow$ GetSynonyms(RK);
$CST \leftarrow CST \cup$ Joining($RK, \{D_h\}$);
return $CST = \{(st_j, freq_j)\}$. ▷ *candidate semantic tags*

3.2 Ranking the Candidates

The second module of Ostia organizes, for each dimension D_h, the candidate semantic tags CST_h into a *candidate tree* $CT_h \subseteq D_h$, and then computes the overall Top-m results. Ranking is based on *weights*. The weight w_j of st_j is computed as $w_j = freq_j \cdot util_j$, where $freq_j$ is the frequency of st_j and $util_j$ is

[3] WordNet: `http://wordnet.princeton.edu`.

the so-called *utility* of st_j wrt *all* other candidates $st_i \in CST_h$, defined as:

$$util_j = \sum_{st_i \in CST_h, i \neq j} \frac{\text{len}(st_j \cap st_i)}{MaxP_h} \quad (1)$$

where $\text{len}(st_j \cap st_i)$ is the length of the common (prefix) path between st_j and st_i, whereas $MaxP_h$ is the maximum path length within the dimension D_h. Utility measures the amount of overlap between st_j and all other st_i's, and aims to score higher: a) longer (i.e., more specific) semantic tags (since for such candidates the degree of overlap with the other candidates is likely to be high), and/or b) candidates occurring in a "dense" part of the candidate tree. On the other hand, the frequency tends to be higher for more generic semantic tags because it is more common to provide generic annotations than specific ones.

Computing all the utilities by directly applying Equation 1 would require $O(N_h^2 \cdot MaxP_h)$ time, with N_h being the cardinality of CST_h. To reduce the computational overhead, we present an equivalent, but more efficient (linear), algorithm. For a semantic tag $st_j = n_0/n_1/\ldots/n_k$, whether st_j is a candidate or not, let us say that the *count* cnt_j of st_j is the number of candidates $st_i \in CST_h$ that contain st_j as a prefix (i.e., of which st_j is an *ancestor*): $cnt_j = \sharp$*candidate semantic tags* st_i *of type* $n_0/\ldots/n_k/\ldots/n_p$, $p \geq k$.

Figure 3 (a) shows an example. For instance, the candidate `a/b/d` has frequency 5 (as given) and count 3, since the number of candidates whose prefix is `a/b/d` is 3, i.e., `a/b/d/g`, `a/b/d/h`, and `a/b/d` itself.

st_j	$freq_j$
a/b/d/g	3
a/b/d	5
a/b/d/h	2
a/b/e	4
a/c/f/i	2

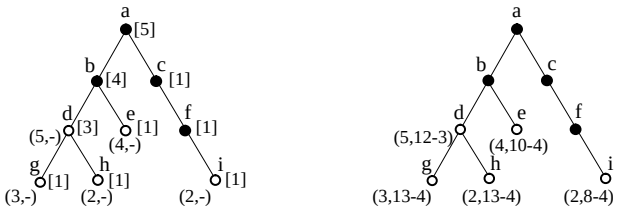


Fig. 3. Candidate tree example (a). Blank circles denote candidate semantic tags (e.g., the one labelled `d` corresponds to the candidate semantic tag `a/b/d`). Close to each candidate st_j, the pair $(freq_j, util_j)$ is shown ($util_j$ is initially undefined), whereas count values $[cnt_i]$ are shown for each node n_i. The tree is completed in (b) with the utility values of the candidates. For clarity of exposition, in this figure we do not normalize utility values by $MaxP_h$.

Theorem 1 *The utility* $util_j$ *of the candidate semantic tag* $st_j = n_0/n_1/\ldots/n_k$ *can be computed as:*

$$util_j = \frac{\sum_{l=0}^{k} cnt_l - len(st_j)}{MaxP_h} = \frac{\sum_{l=0}^{k}(cnt_l - 1)}{MaxP_h} \quad (2)$$

where cnt_l *is the count of the semantic tag* $n_0/n_1/\ldots/n_l$*, ancestor of* st_j*.*

Figure 3 (b) completes the example of Figure 3 (a) showing the utility values of all candidates. For instance, the utility of the semantic tag `a/b/d/g` is $((5 + 4 + 3 + 1) - \text{len}(\texttt{a/b/d/g}))/MaxP_h = (13 - 4)/MaxP_h = 9/MaxP_h$. The same result is obtained from Equation 1, which would compute the utility as $(\text{len}(\texttt{a/b/d/g} \cap \texttt{a/b/d}) + \text{len}(\texttt{a/b/d/g} \cap \texttt{a/b/d/h}) + \text{len}(\texttt{a/b/d/g} \cap \texttt{a/b/e}) + \text{len}(\texttt{a/b/d/h} \cap \texttt{a/c/f/i})/MaxP_h = (3 + 3 + 2 + 1)/MaxP_h = 9/MaxP_h$.

The utilities of all candidates in CST_h can be computed in $O(N_h \cdot MaxP_h)$ time if counts are available. Counts are incrementally obtained while generating the candidate tree CT_h, by adding 1 to the count of a semantic tag st_l whenever a new candidate st_j of which st_l is an ancestor is added to CT_h, as detailed in Algorithm 2.

Algorithm 2 Ostia: Optimal Set of Semantic Tags for Q

Input: CST: candidate semantic tags, m: integer
Output: ST: top-m predicted semantic tags for Q

for all D_h **do**
  $CT_h \leftarrow \emptyset$;
  while $\exists$ a candidate semantic tag $st_j \in CST_h$ **do**
    addCandidateTagToTree($(st_j, freq_j)$, CT_h); ▷ *add candidate to tree*
    for all $n_i \in st_j = n_0/n_1/\ldots/n_k$ **do**
      if n_i is a newly added node in CT_h **then**
        $cnt_i \leftarrow 1$
      else $cnt_i \leftarrow cnt_i + 1$;
  computeUtilities(CT_h); ▷ *compute the utility of all the candidates*
  computeWeights(CT_h, CST_h); ▷ *compute the weight of all the candidates*
  $ST_h \leftarrow \text{Top}(CST_h, m)$; ▷ *optimal set of semantic tags for dimension D_h*
return $ST \leftarrow \text{Top}(\bigcup_{h=1}^{M} ST_h, m)$. ▷ *optimal set of semantic tags*

4 Experimental Results

We have implemented Ostia within our Scenique system, which makes use of the Windsurf library[4] for low-level feature managements (e.g., image segmentation and support for k-NN queries, see [3] for more details). For experiments, we used a dataset of about 100,000 images extracted from the CoPhIR collection [5]. For the dimensions, we imported portions of open-access ontologies from Swoogle[5], for a total of 10 dimensions. The query workload consisted of 50 randomly chosen images. Each query image was assigned a set of semantic tags (3, on the average) by a set of volunteers so as to obtain a ground truth to evaluate the effectiveness of Ostia, which was done by using classical precision (i.e., % of relevant predicted semantic tags) and recall (i.e., % of relevant predicted term with respect to those in the ground truth) metrics. The experiments were performed in the worst-case scenario, where each image $I_i \in \mathcal{DB}$ has no semantic tag yet, i.e., $ST_i = \emptyset$.

Figure 4 shows a sample visual result of Ostia for the picture Q_{912} with associated keywords $K_{912} = \{\texttt{photo}, \texttt{shangai}\}$. As we can observe, the predicted semantic tags (pointed by arrows in the figure), are all relevant for the query. Note that none of them contains keywords in K_{912}.

Figure 5 shows the annotation accuracy of Ostia in term of precision and recall when varying the number of predicted semantic tags m. It can be observed

[4] Windsurf: http://www-db.deis.unibo.it/Windsurf/
[5] Swoogle: http://swoogle.umbc.edu/

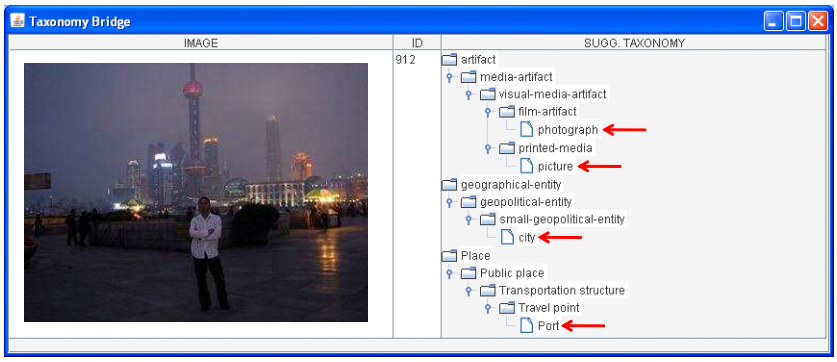
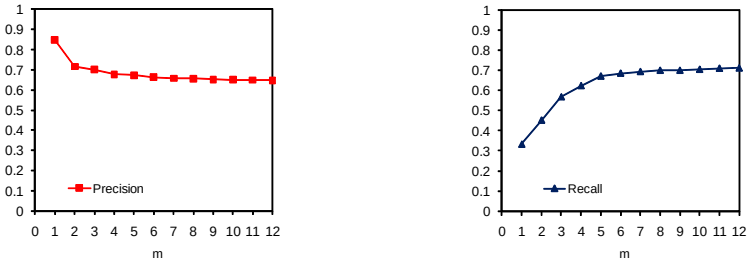


Fig. 4. A visual example of Ostia in action. The semantic tags predicted for the query Q_{912} (with extracted keywords $K_{912} = \{\texttt{photo}, \texttt{shangai}\}$) are pointed by arrows.

that Ostia reaches high level of precision for low values of m (about 85% on the average when $m = 1$) and that it is able to maintain a good quality even for higher m values, by guaranteeing, at the same time, a good level of recall (around 70% for $m \in [6, 12]$).



(a) (b)

Fig. 5. Precision (a) and recall (b) varying the number of predicted semantic tags.

5 Conclusions

In this paper we introduced Ostia, an original algorithm that takes the advantages of both image visual features and keywords, in order to predict for an image a high-quality set of concepts taken from "light-weight" ontologies (or classification hierarchies), here called semantic tags. Ostia can work in a focused way, i.e., predicting semantic tags only for a subset of user-specified dimensions. Further, it can also work even if no keywords are available for a query image, which is the typical case when an image contains no metadata at all, as well as in an incremental way, i.e., by predicting semantic tags for an image with semantic tags (e.g., because a new dimension has been added).

Typically, general-purpose annotation approaches are based on machine learning techniques, that are used to train a set of concept classifiers [8, 6]. The limit

of this approach is that it requires a new classifier to be built from scratch whenever a new class/concept is needed. On the other hand, Ostia does not require a learning phase, thus concepts can be freely added.

Among solutions which uses both visual features and text annotations without pre-defined classes, [7] exploits the query visual features and its *geotags* to derive a set of similar images in the database from which, by means of a frequency-based procedure, geographically relevant tags are predicted. A similar approach is followed in [1], even if not resticted to the geographical case. [6] adds the use of Wordnet to prune uncorrelated tags. However, all these approaches predict free tags only, rather than concepts in a taxonomy as Ostia does.

Future work will deal with the problem of exploiting the hierarchical nature of dimensions and of Wordnet concepts to improve the search of correct synonyms for a given keyword. Further, reasoning on the correlation of predicted semantic tags is an open issue.

References

1. I. Bartolini and P. Ciaccia. Imagination: Accurate Image Annotation Using Link-analysis Techniques. In AMR 2007, pages 32–44.
2. I. Bartolini and P. Ciaccia. Integrating Semantic and Visual Facets for Browsing Digital Photo Collections. In SEBD 2009, pages 65–72.
3. I. Bartolini, P. Ciaccia, and M. Patella. Query Processing Issues in Region-based Image Databases. *Knowledge and Information Systems*, 2010. To appear.
4. M. Bates. Models of Natural Language Understanding. *National Academy of Sciences of the U.S.A*, 92(22):9977–9982, 1995.
5. P. Bolettieri, A. Esuli, F. Falchi, C. Lucchese, R. Perego, T. Piccioli, and F. Rabitti. CoPhIR: a Test Collection for Content-Based Image Retrieval. *CoRR*, abs/0905.4627v2, 2009.
6. R. Datta, W. Ge, J. Li, and J. Z. Wang. Toward Bridging the Annotation-Retrieval Gap in Image Search. *IEEE MultiMedia*, 14(3):24–35, 2007.
7. J. Kleban, E. Moxley, J. Xu, , and B. S. Manjunath. Global Annotation of Georeferenced Photographs. *ACM Conference on Image and Video Retrieval*, 2009.
8. J. Li and J. Z. Wang. Real-time Computerized Annotation of Pictures. In MM 2006, pages 911–920.
9. A. Payne and S. Singh. A Benchmark for Indoor/Outdoor Scene Classification. In ICAPR 2005, pages 711–718.
10. A. Penta, A. Picariello, and L. Tanca. Multimedia Knowledge Management using Ontologies. In MS 2008, pages 24–31.
11. A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-Based Image Retrieval at the End of the Early Years. *IEEE TPAMI*, 22(12):1349–1380, 2000.
12. R. Tye, G. Nathaniel, and N. Mor. Towards Automatic Extraction of Event and Place Semantics from Flickr Tags. In SIGIR 2007, pages 103–110.