Understanding Web Images by Object Relation Network

Na Chen
Dept. of Computer Science
University of Southern California
Los Angeles, CA, USA
nchen@usc.edu

Qian-Yi Zhou
Dept. of Computer Science
University of Southern California
Los Angeles, CA, USA
qianyizh@usc.edu

Viktor K. Prasanna
Ming Hsieh Dept. of Electrical Engineering
University of Southern California
Los Angeles, CA, USA
prasanna@usc.edu

Figure 1: Images and their Object Relation Networks automatically generated by our system

ABSTRACT

This paper presents an automatic method for understanding and interpreting the semantics of unannotated web images. We observe that the relations between objects in an image carry important semantics about the image. To capture and describe such semantics, we propose Object Relation Network (ORN), a graph model representing the most probable meaning of the objects and their relations in an image. Guided and constrained by an ontology, ORN transfers the rich semantics in the ontology to image objects and the relations between them, while maintaining semantic consistency (e.g., a soccer player can kick a soccer ball, but cannot ride it). We present an automatic system which takes a raw image as input and creates an ORN based on image visual appearance and the guide ontology. We demonstrate various useful web applications enabled by ORNs, such as automatic image tagging, automatic image description generation, and image search by image.

Categories and Subject Descriptors
I.4 [Image Processing and Computer Vision]: Image Representation, Scene Analysis, Applications

Keywords
Image understanding, image semantics, ontology, detection

1. INTRODUCTION

Understanding the semantics of web images has been a critical component in many web applications, such as automatic web image interpretation and web image search. Manual annotation, particularly tagging, has been considered as a reliable source of image semantics due to its human origins. Manual annotation can yet be very time-consuming and expensive when dealing with web-scale image data. Advances in Semantic Web have made ontology another useful source for describing image semantics (e.g., [23]). Ontology builds a formal and explicit representation of semantic hierarchies for the concepts and their relationships in images, and allows reasoning to derive implicit knowledge. However, the gap between ontological semantics and image visual appearance is still a hindrance towards automated ontology-driven image annotation. With the rapid growth of image resources on the world-wide-web, vast amount of images with no metadata have emerged. Thus automatically understanding raw images solely based on their visual appearance becomes an important yet challenging problem.

Advances in computer vision have offered computers an eye to see the objects in images. In particular, object detection [8] can automatically detect what is in the image and where it is. For example, given Fig. 2(a) as the input image to object detectors, Fig. 2(b) shows the detected objects and their bounding boxes. However, current detection techniques have two main limitations. First, detection is limited to isolated objects and cannot see through the relations between them. Second, only generic objects are detected; detection quality can drop significantly when detectors attempt to assign more specific meaning to these objects. For instance, detectors successfully detected one person and one ball in Fig. 2(b), but cannot further tell whether the person is throwing or holding the ball, or whether the person is a basketball player or a soccer player.

This paper presents an automatic system for understanding web images with no metadata, by taking advantages of both ontology and object detection. Given a raw image as input, our system adopts object detection in pre-processing to find generic objects in the image; and em-
ploys a guide ontology as a semantic source of background knowledge. We propose Object Relation Network (ORN) to transfer rich semantics in the guide ontology to the detected objects and their relations. In particular, ORN is defined as a graph model representing the most probable ontological class assignments for the objects and their relations. Our method automatically generates ORN for an image by solving an energy optimization problem over a directed graphical model. The output ORN can be regarded as an instantiation of the guide ontology with respect to the input image. Fig. 1 illustrates three web images and their ORNs automatically generated by our system.

Object Relation Networks can be applied to many web applications that need automatic image understanding. In particular, we demonstrate three typical applications:

**Automatic image tagging:** With a few simple inference rules, ORNs can automatically produce informative tags describing entities, actions, and even scenes in images.

**Automatic image description generation:** Natural language description of an image can be automatically generated based on its ORN, using a simple template based approach.

**Image search by image:** Given a query image, the objective is to find images semantically-similar to the query image from a image library. We show that the distance between ORN graphs is an effective measurement of image semantic similarity. Search results consist of images with ORNs that are close to the query image’s ORN, ranked by ORN distances.

The main contributions of this paper include:

1. We propose and exploit Object Relation Network towards automatic web image understanding. ORN is an intuitive and expressive graph model in representing the semantics of web images. It presents the most probable meaning of image objects and their relations, while maintaining the semantic consistency through the network.

2. We combine ontological knowledge and image visual features into a probabilistic graphical model. By solving an optimization problem on this graphical model, our method automatically transfers rich semantics in the ontology to a raw image, generating an ORN in which isolated objects detected from the image are connected through meaningful relations.

3. We propose and demonstrate three application scenarios that can benefit from ORNs: automatic image tagging, automatic image description generation, and image search by image.

## 2. RELATED WORK

Our work is related to the following research areas.

**Image understanding with keywords and text:** Some research achievements have been made in the web community towards understanding web image semantics with keywords and text, such as tag recommendation, tag ranking and transfer learning from text to images. Tag recommendation [24, 29] enriches the semantics carried by existing tags by suggesting similar tags. Tag ranking [15, 28] identifies the most relevant semantics among existing tags. Transfer learning from text to images [19] builds a semantic linkage between text and image based on their co-occurrence. These methods all require images to have meaningful initial tags or relevant surrounding text, and thus do not work for untagged images or images with irrelevant text surrounded.

**Image understanding using visual appearance:** Computer vision community has made great progress in automatically identifying static objects in images, also known as object detection. The PASCAL visual object classes challenge (VOC) is an annual competition to evaluate the performance of detection approaches. For instance, in the VOC2011 challenge [5], detectors are required to detect twenty object classes from over 10,000 flickr images. These efforts have made object detectors a robust and practical tool for extracting generic objects from images (e.g., [8]). On the other hand, detection and segmentation are usually localized operations and thus lose information about the global structure of an image. Therefore, contextual information is introduced to connect localized operations and the global structure. In particular, researchers implicitly or explicitly introduce a probabilistic graphical model to organize pixels, regions, or detected objects. The probabilistic graphical model can be a hierarchical structure [27, 14], a directed graphical model [26], or a conditional random field [10, 20, 11, 12]. These methods are similar in spirit to our method, however, there are two key differences: (1) we introduce ontology to provide semantics for both relations and objects, while previous research (even with an ontology [18]) focuses on spatial relationships, such as on-top-of, beside, which are insufficient to satisfy the semantic demand of web applications; (2) previous research usually focuses on improving local operations or providing a general description for the entire scene, they do not explicitly reveal the semantic relations between objects thus are less informative than our Object Relation Network model.

**Ontology-aided image annotation:** A number of annotation ontologies (e.g., [23, 21]) have been proposed to provide description templates for image and video annotation. Concept ontology [6] characterizes the inter-concept correlations to help image classification. Lexical ontologies, particularly the WordNet [7] ontology, describe the semantic hierarchies of words. WordNet groups words into sets of synonyms and records different semantic relations between them, such as antonymy, hypernymy and meronymy. The WordNet ontology has been used to: (1) improve or extend existing annotations of an image [25, 3], (2) provide knowledge about the relationships between object classes for object category recognition [17], and (3) organize the structure of image database [4]. Different from the prior work, we exploit ontologies to provide background knowledge for automatic image understanding. In particular, the key difference between our guide ontology and the ontology in [17] is that our guide ontology contains semantic hierarchies of both object classes and relation classes, and supports various semantic constraints.

## 3. SYSTEM OVERVIEW

An overview of our system is illustrated in Fig. 2. Taking an unannotated image (Fig. 2(a)) as input, our system first employs a number of object detectors to detect generic objects from the input image (Fig. 2(b)). The guide ontology (Fig. 2(c), detailed in Section 4) contains useful background knowledge such as semantic hierarchies and constraints related to the detected objects and their relations. Our system then constructs a directed graphical model as a princi-
In general, guide ontologies should have three layers. The root layer contains three general classes, Object, OO-Relation, and Object Collection, denoting the class of image objects, the class of binary relations between image objects, and the class of image object collections, respectively. The detection layer contains classes of the generic objects that can be detected by the object detectors in our system. Each of these class is an immediate subclass of Object, and corresponds to a generic object (e.g., person and ball). The semantic knowledge layer contains background knowledge about the semantic hierarchies of object classes and relation classes, and the constraints on relation classes. Each object class at this layer must have a superclass in the detection layer, while each relation class must be a subclass of OO-Relation.

Conceptually, any ontology regarding the detectable objects and their relations can be adapted into our system as part of the semantic knowledge layer, ranging from a tiny ontology which contains only one relation class with a domain and a range class, to large ontologies such as WordNet [7]. However, ontologies with more hierarchical information and more restrictions are always preferred since they carry more semantic information. Our system supports four typical types of background knowledge in the guide ontology:

- **Subsumption** is a relationship where one class is a subclass of another, denoted as $A \subseteq B$. E.g., Basketball Player is a subclass of Athlete.
- **Domain/range constraints** assert the domain or range object class of a relation class, denoted as $domain(C)$ and $range(C)$. E.g., in Fig. 2(c), the domain and the range of Kick must be Soccer Player and Soccer Ball respectively.
- **Cardinality constraints** limit the maximum number of relations of a certain relation class that an object can have, where the object’s class is a domain/range of

4. **GUIDE ONTOLOGY**

The source of semantics in our system is a guide ontology. It provides useful background knowledge about image objects and their relations. An example guide ontology is shown in Fig. 2(c).

In general, guide ontologies should have three layers. The root layer contains three general classes, Object, OO-Relation, and Object Collection, denoting the class of image objects, the class of binary relations between image objects, and the class of image object collections, respectively. The detection layer contains classes of the generic objects that can be detected by the object detectors in our system. Each of these class is an immediate subclass of Object, and corresponds to a generic object (e.g., person and ball). The semantic knowledge layer contains background knowledge about the semantic hierarchies of object classes and relation classes, and the constraints on relation classes. Each object class at this layer must have a superclass in the detection layer, while each relation class must be a subclass of OO-Relation.

Conceptually, any ontology regarding the detectable objects and their relations can be adapted into our system as part of the semantic knowledge layer, ranging from a tiny ontology which contains only one relation class with a domain and a range class, to large ontologies such as WordNet [7]. However, ontologies with more hierarchical information and more restrictions are always preferred since they carry more semantic information. Our system supports four typical types of background knowledge in the guide ontology:

- Subsumption is a relationship where one class is a subclass of another, denoted as $A \subseteq B$. E.g., Basketball Player is a subclass of Athlete.
- Domain/range constraints assert the domain or range object class of a relation class, denoted as $domain(C)$ and $range(C)$. E.g., in Fig. 2(c), the domain and the range of Kick must be Soccer Player and Soccer Ball respectively.
- Cardinality constraints limit the maximum number of relations of a certain relation class that an object can have, where the object’s class is a domain/range of
the relation class. E.g., in Fig. 2(c), Basketball Player \( \xrightarrow{\leq 1} \) Throw means that a Basketball Player can have at most one Throw relation.

- **Collection** refers to a set of image objects belonging to the same object class, denote as \( \text{collection}(C) \).

5. DIRECTED GRAPHICAL MODEL

The core of our system is a directed graphical model \( G = (V, E) \). It is a primitive relation network connecting the detected objects through relations. In particular, given a set of detected objects \( \{O_i\} \) in the input image, we create one object node \( o_i \) for each object \( O_i \), and one relation node \( r_{i,j} \) for each object pair \( <O_i, O_j> \) that has a corresponding relation class in the detection layer of the guide ontology (e.g., object pair \(<\text{person1}, \text{ball1}>\) corresponding to class \( \text{P-B Relation} \)), indicating that the two objects have potential relations. For each relation node \( r_{i,j} \), we create two directed edges \((o_i, r_{i,j})\) and \((r_{i,j}, o_j)\). These nodes and edges form the basic structure of \( G \).

We now consider a labeling problem over the node set \( V \) of graph \( G \): for each node \( v \in V \), we label it with a class assignment from the subclass tree rooted at the generic class corresponding to \( v \). In particular, we denote the potential class assignments for object node \( o_i \) as \( C(o_i) = \{C_o \subseteq C_g(O_i)\} \), where \( C_g(O_i) \) is the generic class of object \( O_i \) (e.g., Person for object person1). Similarly, the set of potential class assignments for relation node \( r_{i,j} \) is defined as \( C(r_{i,j}) = \{C_r | C_r \subseteq C_g(O_i, O_j)\} \), where \( C_g(O_i, O_j) \) is the corresponding relation class in the detection layer (e.g., P-B Relation). A labeling \( L : \{v \sim C(v)\} \) is feasible when we have \( C(v) \in C_i(v) \) for each node \( v \in V \).

The best feasible labeling \( L_{\text{optimal}} \) is required to (1) satisfy the ontology constraints, (2) be as informative as possible, and (3) maximize the probability of the class assignment on each node regarding visual appearance. We predict \( L_{\text{optimal}} \) by minimizing an energy function \( E \) over labeling \( L \) with respect to an image \( Img \) and a guide ontology \( Ont \):

\[
E(L) = E_v(L; Ont) + E_e(L; Ont) + E_r(L; Img) \tag{1}
\]

representing the sum of the constraint energy, the informative energy, and the visual energy; which are detailed in the following subsections respectively.

5.1 Ontology based energy

We define energy functions based on background knowledge in the guide ontology \( Ont \).

**Domain/range constraints** restrict the potential class assignments of a relation’s domain or range. Thus, we define a domain-constraint energy for each edge \( e = \{o_i, r_{i,j}\} \) and a range-constraint energy for each edge \( e = \{r_{i,j}, o_j\} \):

\[
E^D_L(o_i \sim C_o, r_{i,j} \sim C_r) = \begin{cases} 
0 & \text{if } C_o \subseteq \text{domain}(C_r) \\
\infty & \text{otherwise}
\end{cases}
\]

\[
E^R_L(r_{i,j} \sim C_r, o_j \sim C_o) = \begin{cases} 
0 & \text{if } C_o \subseteq \text{range}(C_r) \\
\infty & \text{otherwise}
\end{cases}
\]

Intuitively, they add strong penalty to the energy function when any of the domain/range constraints is violated.

**Cardinality constraints** restrict the number of instances of a certain relation class that an object can take. We are particularly interested in cardinality constraint of 1 since it is the most common case in practice. In order to handle this type of constraints, we add additional edges as shown in Fig. 3. In particular, if a relation class \( C_r \) has a cardinality constraint being 1 with its domain (or range), we create an edge between any relation node pair \((r_{i,j}, r_{k,l})\) or \((r_{i,j}, r_{k,l})\) when dealing with range) in which both nodes have a same domain node \( o_i \) (or range node \( o_j \)) and both nodes have potential of being labeled with relation class \( C_r \). A cardinality constraint energy is defined on these additional edges:

\[
E^C_L(o_i \sim C_o \rightarrow C_1, r_{i,j} \sim C_2) = \begin{cases} 
\infty & \text{if } C_1 = C_2 = C_r \\
0 & \text{otherwise}
\end{cases}
\]

Intuitively, they penalize the energy function when two relations are assigned as \( C_r \) and have the same domain object (or range object).

**Depth information**, defined as the depth of a class assignment in the subclass tree rooted at its generic class. Intuitively, we prefer deep class assignments which are more specific and more informative. In contrast, general class assignments with small depth should be penalized since they are less informative and thus may be of less interest to the user. In the extremely general case where generic object classes are assigned to the object nodes and \( OO-\text{Relation} \) is assigned to all the relation nodes, the labeling is feasible but should be avoided since little information is revealed.

Therefore, we add an energy function for each node \( o_i \) or \( r_{i,j} \) concerning depth information:

\[
E^D_L(o_i \sim C_o) = -\omega_{\text{dep}} \cdot \text{depth}(C_o)
\]

\[
E^R_L(r_{i,j} \sim C_r) = -\omega_{\text{dep}} \cdot \text{depth}(C_r)
\]

**Collection** refers to a set of object nodes with the same class assignment. Intuitively, we prefer collections with larger size as they tend to group more objects in the same image to the same class. For example, in Fig. 4, when the two persons in the front are labeled with \( \text{Soccer Player} \) due to the strong observation that they may \( \text{Kick a Soccer Ball} \) how to make observations from visual features is detailed in Sec 5.2), it is quite natural to label the third person with \( \text{Soccer Player} \) as well since the three of them will form a relatively larger \( \text{Soccer Players Collection} \). In addition, we bonus collections that are deeper in the ontology, e.g., we prefer \( \text{Soccer Players Collection} \) to \( \text{Person Collection} \). To integrate collection information into our energy minimization framework is a bit complicated since we do not explicitly have graph nodes for
collections. Therefore, we add edges between object nodes \((o_i, o_j)\) when they belong to the same generic object class that has the potential to form a collection (e.g., Fig. 4 right), and define an energy function for each such edge:

\[
E^\text{Col}_{o_i \sim C_1, o_j \sim C_2} = \begin{cases} 
-\omega_{\text{col}} \times \text{depth}(\text{collection}(C_o)) & \text{if } C_1 = C_2 = C_o \\
0 & \text{otherwise}
\end{cases}
\]

where \(\omega_{\text{col}}\) is a weight, and \(\frac{2}{\text{depth}(C_o)}\) is a normalization factor with \(N\) representing the number of object nodes that can be potentially labeled with \(C_o\).

Finally, the ontology based constraint energy \(E_c(L; \text{Ont})\) and informative energy \(E_i(L; \text{Ont})\) are the sum of these energy functions:

\[
E_c(L; \text{Ont}) = \sum_{(o_i, r_{i,j})} E^D_c + \sum_{(r_{i,j}, o_j)} E^R_c + \sum_{(r_{i,j}, r_{k,j})} E^\text{Card}_c + \sum_{(r_{i,j}, r_{k,j})} E^\text{Card}_c(2)
\]

\[
E_i(L; \text{Ont}) = \sum_{o_i} E^O_i + \sum_{r_{i,j}} E^R_i + \sum_{(o_i, o_j)} E^\text{Col}_i (3)
\]

### 5.2 Visual feature based energy

Besides background knowledge from the ontology, we believe that visual appearance of objects can give us additional information in determining class assignments. E.g., a Ball with white color is more likely to be a Soccer Ball, while the relation between two spatially close objects are more probable to be \textit{Interact} than \textit{Non-interact}. Thus, we define visual feature based energy functions for object nodes and relation nodes respectively.

**Visual feature based energy on object nodes:** for each object node \(o_i\), we collect a set of visual features \(F_o(o_i)\) of the detected object \(O_i\) in the input image, and calculate a probability distribution over potential assignment set \(C(\alpha_i)\) based on \(F_o(o_i)\). Intuitively, the conditional probability function \(P(o_i \sim C_0 | F_o(o_i))\) denotes the probability of \(o_i\) assigned as class \(C_0\) when \(F_o(o_i)\) is observed from the image. Thus, we define the visual feature based energy on an object node as:

\[
E^V_o(o_i \sim C_0) = -\omega_{\text{obj}} P(o_i \sim C_0 | F_o(o_i))
\]

We choose eight visual features of \(O_i\) to form feature set \(F_o(O_i)\), including: width and height of \(O_i\)'s bounding box (which is part of the output from detectors); the average of H.S.V values from the HSV color space; and the standard deviation of H.S.V.

Given these eight feature values on an object node \(o_i\), a probability distribution over the potential assignment set \(C(\alpha_i)\) is estimated, which satisfies:

\[
\sum_{C \in C(\alpha_i)} P(o_i \sim C | F_o(o_i)) = P(o_i \sim C_0 | F_o(o_i)) = 1
\]

where \(C_0(o_i)\) is the generic class of \(O_i\), and \(o_i \subseteq C_0(o_i)\) is the notation for “\(o_i\) is assigned as a subclass of \(C_0(o_i)\)”.

We take advantage of the hierarchical structure of the subclass tree rooted at \(C_0(o_i)\), and compute the probability distribution in a top-down manner. Assume \(P(o_i \subseteq C_0(o_i) | F_o(o_i))\) is known for certain object class \(C_o\); if \(C_o\) is a leaf node in the ontology (i.e., \(C_o\) has no subclass), we have \(P(o_i \sim C | F_o(o_i)) = P(o_i \subseteq C_0(o_i) | F_o(o_i))\); otherwise, given \(C_o\)’s immediate subclass set \(I(C_o)\), we have a propagation equation:

\[
P(o_i \subseteq C_0(o_i) | F_o(o_i)) = P(o_i \sim C_0 | F_o(o_i))
\]

\[
+ \sum_{C_0 \in I(C_o)} P(o_i \subseteq C_0 | F_o(o_i))
\]

We can view the right-hand side of this equation from the perspective of multi-class classification: given conditions \(o_i \subseteq C_0\) and \(F_o(o_i)\), the assignment of \(o_i\) falls into \(|I(C_o)| + 1\) categories: \(o_i \sim C_0\), or \(o_i \subseteq C_0\) where \(C_k \in I(C_o), k = 1, \ldots, |I(C_o)|\). Thus we can train a multi-class classifier (based on object visual features) to assign a classification score for each category, and apply the calibration method proposed in [30] to transform these scores into a probability distribution over these \(|I(C_o)| + 1\) categories. Multiplied by the prior \(P(o_i \subseteq C_0(o_i) | F_o(o_i))\), this probability distribution determines the probability functions on the right-hand side of Eqn.(4). Thus, the probabilities recursively propagate from the root class down to the entire subclass tree, as demonstrated in Fig. 5.

In order to train a classifier for each non-leaf object class \(C_o\), we collect a set of objects \(O_{\text{train}}\) belonging to class \(C_o\) from our training images with ground truth labeling, and calculate their feature sets. The training samples are later split into \(|I(C_o)| + 1\) categories regarding their labeling: assigned as \(C_o\), or belonging to one of \(C_o\)’s immediate subclasses. We follow the suggestions in [30] to train one-against-all SVM classifiers using a radial basis function as kernel [22] for each category, apply isotonic regression (PAV [1]) to calibrate classifier scores, and normalize the probability estimates to make them sum to 1. This training process is made for every non-leaf object class once and for all.

**Visual feature based energy on relation nodes** can be handled in a similar manner to that on object nodes. The only difference is the feature set \(F_r(O_i, O_j)\). As for relations, the relative spatial information is most important. Therefore, \(F_r(O_i, O_j)\) contains eight features of object pair \(<O_i, O_j>\): the width, height and center (both \(x\) and \(y\) coordinates) of \(O_i\)’s bounding box; and the width, height and center of \(O_j\)’s bounding box.

Similarly, training samples are collected and classifiers are trained for each non-leaf relation class \(C_r\) in the ontology. With these classifiers, probabilities propagate from each generic relation class to its entire subclass tree to form a distribution over the potential assignment set \(C(r_{i,j})\). The visual feature based energy on \(r_{i,j}\) is defined as:

\[
E^R_{r_{i,j} \sim C_r} = -\omega_{\text{rel}} P(r_{i,j} \sim C_r | F_r(O_i, O_j))
\]
3. On each node, we sort the potential class assignments by their visual feature based probabilities in descending order. Class assignments with large probabilities are searched first, and those with very small probabilities (empirically, < 0.1) are only searched when no other assignments are available.

3. We finally drop some meaningless relation nodes (in particular, Non-interact), together with the edges touching them.

After these steps, an intuitive and expressive ORN is automatically created from our system to interpret the semantics of the objects and their relations in the input image. Examples are shown in Fig. 1 and Fig. 9.

6. EXPERIMENTAL RESULTS

6.1 Energy function evaluation

We first demonstrate how visual feature based energy functions work together with ontology based energy functions, using the example in Fig. 6. We observe that the probability distributions shown in the middle tend to give a good estimation for each node, i.e., provide a relatively high probability for the true labeling. But there is no guarantee that the probability of the true labeling is always the highest (e.g., Ball1 has a higher probability of being assigned as Ball than Basketball, highlighted in red). By combining the energy functions together, the ontology constraints provide a strict frame to restrict the possible labeling over the entire graph by penalizing inappropriate assignments (e.g., Basketball Player Throw Ball, given that the range of relation Throw is limited to Basketball). Probabilities are organized into a tightly interrelated network which in return improves the prediction for each single node (e.g., in the labeling with minimal energy, Ball1 is correctly assigned as Basketball).

To quantitatively evaluate the energy functions, we collect 1,200 images from ImageNet [4] from category soccer, basketball and ball. A person detector [9] and a ball detector using Hough Circle Transform in OpenCV [2] are applied on the entire image set to detect persons and balls. The detected objects and the relations between them are manually labeled with classes from the guide ontology in Fig. 2(c). We then randomly select 600 images as training data, and use the rest as test data. Three different scenarios are compared: (1) using only visual feature based energy, (2) using both visual feature based energy and ontology constraints, and (3) using the complete energy function \( E(L) \) in Eqn. 1. Our system minimizes the energy cost in each of the scenarios, and calculates the error rate by comparing the system output with ground truth. As Fig. 8 and Table 1 suggest, ontology-based energy transfers background knowledge from the guide ontology to the relation network, and thus significantly improves the quality of class assignments.
Figure 7: The ontology we use for system evaluation. Constraints are only shown in the root layer for clarity.

Figure 8: Error rate of class assignments under three different scenarios.

Table 1: Evaluation results of the energy functions. The first columns contains data for Fig. 8. The last two columns show the gain in accuracy by using the complete energy function \( E(L) \), where \( k \) is the number of detected objects.

<table>
<thead>
<tr>
<th>Generic class</th>
<th>Using ( E_v )</th>
<th>Using ( E_v + E_e )</th>
<th>Using ( E(L) )</th>
<th>Gain</th>
<th>Gain (k&gt;3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Person</td>
<td>0.5188</td>
<td>0.4644</td>
<td>0.3766</td>
<td>0.1423</td>
<td>0.1783</td>
</tr>
<tr>
<td>Ball</td>
<td>0.2967</td>
<td>0.2693</td>
<td>0.2640</td>
<td>0.0267</td>
<td>0.0571</td>
</tr>
<tr>
<td>P-B Rel.</td>
<td>0.3222</td>
<td>0.2887</td>
<td>0.2259</td>
<td>0.0692</td>
<td>0.0775</td>
</tr>
</tbody>
</table>

Table 2: Human evaluation of ORN and detection: possible score ranges from 5(perfect) to 1(failure). \( k \) is the number of detected objects.

<table>
<thead>
<tr>
<th>ORN score</th>
<th>( k = 1 )</th>
<th>( k = 2 )</th>
<th>( k = 3 )</th>
<th>( k &gt; 3 )</th>
<th>overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detection score</td>
<td>4.31</td>
<td>3.94</td>
<td>3.10</td>
<td>3.38</td>
<td>3.65</td>
</tr>
</tbody>
</table>

6.2 System evaluation

To further evaluate the robustness and generality of our system, we adapt a more complicated guide ontology (Fig. 7) into the system. The detection layer contains 6 generic object classes: Person, Horse, Motorbike, Chair, Bicycle, and Ball, while the semantic layer contains simplified semantic hierarchies from the WordNet ontology [7]. Moreover, we extend our image set with images from VOC2011 [5] containing over 28,000 web images. We randomly choose 2,000 images that have at least one generic object, manually label ground truth class assignments for objects and relations, and use them to train the visual feature based classifiers and the weight set \((\omega_{\text{dep}}, \omega_{\text{col}}, \omega_{\text{obj}}, \omega_{\text{col}})\). We adopt the detectors in [9, 2] to perform object detection.

Time complexity: The most time-consuming operation of our system is detection, which usually takes around one minute per test image. After this pre-processing, our system automatically creates an ORN for each image within a second. All experiments are run on a laptop with Intel i-7 CPU 1.60GHz and 6GB memory.

Qualitative results: Most of our ORNs are quite good. Example results are shown in Fig. 9. The “Good” ORNs in the first four columns successfully interpret the semantics of the objects and their relations. We also demonstrate some “bad” examples in the last column. Note that the “bad” results are usually caused by detection errors (e.g., the top image has a false alarm from the person detector while the rest two images both miss certain objects). Nevertheless, the “bad” ORNs still interpret reasonable image semantics.

Human evaluation: We perform human judgement on the entire test data set. Scores on a scale of 5 (perfect) to 1 (failure) are given by human judges to reflect the quality of the ORNs, shown in Table 2. First, we notice that the ORNs are quite satisfactory as the overall score is 3.65. Second, ORN scores for images of a single object are relatively high because the detection is reliable when \( k = 1 \). With the number of objects increasing, the relation network becomes larger and thus more ontology knowledge is brought into the optimization process. The quality of ORN keeps improving despite the quality drop of detection.

7. APPLICATIONS

We present three typical web applications based on the rich semantic information carried by ORNs, detailed as follows.

7.1 Automatic image tagging

We develop an automatic image tagging approach by combining ORNs and a set of inference rules. Given a raw image as input, our system automatically generates its ORN which contains semantic information for the objects, relations, and collections. Thus, we directly output the ontological class assignments in the ORN as tags regarding entities, actions, and entity groups. In addition, with a few simple rules, implicit semantics about the global scene can also be easily inferred from the ORN and translated into tags. Table 3 shows some example inference rules. Results from our method and a reference approach ALIPR [13] are illustrated in the third row of Fig. 10. Note that even with imperfect ORNs (the 5th and 6th image), our approach is still capable of producing relevant tags.
Figure 9: Object Relation Networks are automatically generated from our system. “Good” results are illustrated in the first four columns while the last column shows some “bad” results.

Table 3: Example rules for inferencing implicit knowledge from ORNs

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( \exists x \text{SoccerPlayerCollection}(x) \land \exists y \text{SoccerBall}(y) \land \exists z \text{SoccerPlayer}(z) \land (\text{kick}(z, y) \lor \text{head}(z, y)) \land \text{Tag}(t) \rightarrow t = \text{&quot;soccer game&quot;} )</td>
</tr>
<tr>
<td>2</td>
<td>( \exists x \text{CyclistCollection}(x) \land \exists y \text{Cyclist}(y) \land \exists z \text{Bicycle}(z) \land \text{ride}(y, z) \land \text{Tag}(t) \rightarrow t = \text{&quot;bicycle race&quot;} )</td>
</tr>
<tr>
<td>3</td>
<td>( \exists x \text{BasketballPlayerCollection}(x) \land \exists y \text{Basketball}(y) \land \exists z \text{BasketballPlayer}(z) \land (\text{throw}(z, y) \lor \text{hold}(z, y)) \land \text{Tag}(t) \rightarrow t = \text{&quot;basketball game&quot;} )</td>
</tr>
</tbody>
</table>

7.2 Automatic image description generation

Natural language generation for images is an open research problem. We propose to exploit ORNs to automatically generate natural language descriptions for images. We extend our automatic tagging approach by employing a simple template based model (inspired by [11]) to transform tags into concise natural language sentences. In particular, the image descriptions begin with a sentence regarding the global scene, followed by another sentence enumerating the entities (and entity groups if there is any) in the image. The last few sentences are derived from relation nodes in the ORN with domain and range information. Examples are shown in the last row of Fig. 10.

7.3 Image search by image

The key in image search by image is the similarity measurement between two images. Since ORN is a graph model that carries informative semantics about an image, the graph distance between ORNs can serve as an effective measurement of the semantic similarity between images. Given that ORN is an ontology instantiation, we employ the ontology distance measurement in [16] to compute ORN distances. In particular, we first pre-compute the ORNs for images in our image library which contains over 30,000 images. Then for each query image, we automatically generate its ORN, and retrieve images with the most similar ORNs from the image library. The result images are sorted by ORN distances. Fig. 11 illustrates several search results of our approach. Search results from Google Image Search by Image are also included for reference.

8. CONCLUSION

We presented Object Relation Network (ORN) to carry semantic information for web images. By solving an optimization problem, ORN was automatically created from a graphical model to represent the most probable and informative ontological class assignments for objects detected from the image and their relations, while maintaining semantic consistency. Benefiting from the strong semantic expressiveness of ORN, we proposed automatic solutions for three typical yet challenging image understanding problems. Our experiments showed the effectiveness and robustness of our system.

9. ACKNOWLEDGMENTS

We acknowledge support from NSF grant CCF-1048311.

10. REFERENCES

Figure 10: Tags and natural language descriptions automatically generated from our ORN-based approaches. Annotation results from the ALIPR system (http://alipr.com/) are also showed for reference.


Figure 11: Image search results of our approach and Google Image Search by Image (http://images.google.com/)

semantic knowledge propagation from text corpus to web images. In WWW, 2011.