Learning the Visual Interpretation of Sentences

C. Lawrence Zitnick
Microsoft Research
larryz@microsoft.com

Devi Parikh
Virginia Tech
parikh@vt.edu

Lucy Vanderwende
Microsoft Research
Lucy.Vanderwende@microsoft.com

Abstract

Sentences that describe visual scenes contain a wide variety of information pertaining to the presence of objects, their attributes and their spatial relations. In this paper we learn the visual features that correspond to semantic phrases derived from sentences. Specifically, we extract predicate tuples that contain two nouns and a relation. The relation may take several forms, such as a verb, preposition, adjective or their combination. We model a scene using a Conditional Random Field (CRF) formulation where each node corresponds to an object, and the edges to their relations. We determine the potentials of the CRF using the tuples extracted from the sentences. We generate *novel scenes depicting the sentences' visual meaning by sampling from the CRF. The CRF is also used to score a set of scenes for a text-based image retrieval task. Our results show we can generate (retrieve) scenes that convey the desired semantic meaning, even when scenes (queries) are described by multiple sentences. Significant improvement is found over several baseline approaches.*

1. Introduction

Learning the relation of language to its visual incarnation remains a challenging and fundamental problem in computer vision. Both text and image corpora offer substantial amounts of information about our physical world. Relating the information between these domains may improve applications in both and lead to new applications. For instance, image search is commonly performed using text as input. As the fields of Natural Language Processing (NLP) and computer vision progress we may move towards more descriptive and intuitive sentence-based image search.

Recently there has been significant work in relating images to their sentence-based semantic descriptions. This process may be studied in either direction. That is, an image may be given as input and a sentence produced [8, 1, 13, 25, 38, 19], or a scene or animation may be generated from a sentence description [27, 6, 17, 14]. For the later, the quality of the generated scenes may be studied to determine whether the meaning of the sentence was correctly interpreted. A common approach to both of these problems is to manually define the visual interpretation of various semantic phrases. For instance, what does it mean for an object to be "next to" or "above" another object [6, 19]. Recently, Sadeghi *et al.* [31] proposed a novel approach to discovering the visual meaning of semantic phrases using training datasets created from text-based image search.

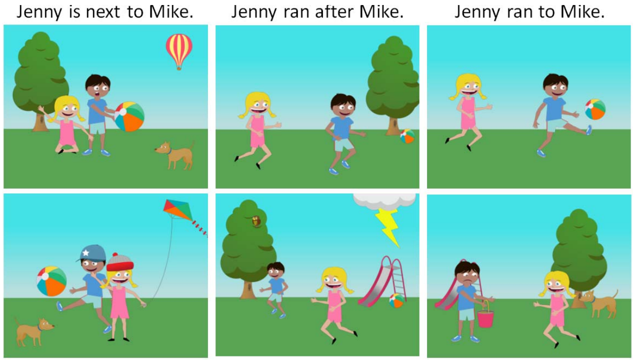


Figure 1: Three example sentences and scenes. Notice how subtle changes in the wording of the sentences leads to different visual interpretations.

In this paper, we study the problem of visually interpreting sentences. We demonstrate the effectiveness of our approach by both generating novel scenes from sentence descriptions, and by enabling sentence-based search of abstract scenes [41]. However, unlike previous papers [6, 19] we automatically discover the relation between semantic and visual information. That is, we not only learn the mapping of nouns to the occurrence of various objects, but also the visual meaning of many verbs, prepositions and adjectives. We explore this problem within the methodology of Zitnick and Parikh [41], which proposed the use of abstract scenes generated from clip art to study semantic scene understanding, Figure 1. The use of abstract scenes over real images provides us with two main advantages. First, by construction we know the visual arrangement and attributes of the objects in the scene, but not their semantic meaning. This allows us to focus on the core problem of semantic scene understanding, while avoiding problems that arise with the use of noisy automatic object and attribute detectors in real images. Second, while real image datasets may be quite large, they contain a very diverse set of scenes resulting in a sparse sampling of many semantic concepts [15, 29, 8, 36]. Using abstract scenes we may densely sample to learn subtle nuances in semantic meaning. Consider the examples in Figure 1. While the sentences "Jenny is next to Mike", "Jenny ran after Mike" and "Jenny ran to Mike" are similar, each has distinctly different visual interpretations. With densely sampled training data we may learn that "ran after" implies Mike also has a running pose, while "run to" does not. Similarly, we could learn that "next to" does not imply that Jenny is facing Mike, while "ran after" and "ran to" do.

We conjecture the information learned on abstract scenes will be applicable to real images. The dataset created by [41] was created with the intent to represent real-world scenes that contain a diverse set of subtle relations. The approach may be applied to a variety of scenarios by varying the type and realism of the clip art used. Specifically, the dataset used contains 10,000 images of children playing outside. For each scene, we gathered two sets of three sentences describing different aspects of the scene. The result is a dataset containing 60,000 sentences that is publicly available on the authors' website. We only use visual features that may be realistically extracted from real images. These include object [12], pose [37], facial expression [11], and attribute [9, 4, 26] detectors.

Our approach models a scene using a Conditional Random Field (CRF) with each node representing a different object. Unary potentials model the position, occurrence and attributes of an object. Pairwise potentials model the co-occurrence and relative position of objects. For each sentence, we extract a set of predicate tuples containing a primary object, a relation, and secondary object. We use the predicate tuples and visual features (trivially) extracted from the training scenes to determine the CRF's unary and pairwise potentials. Given a novel sentence, a scene depicting the meaning of the sentence may be generated by sampling from the CRF. We show several intermediate results that demonstrate that our approach learns an intuitively correct interpretation of semantic relations. Results show that our approach can generate (and retrieve) scenes that convey the desired semantic meaning, even when scenes are described by multiple sentences. Significant improvement is found over several strong baseline approaches.

2. Related work

Images to sentences: Several works have looked at the task of annotating images with tags, be it nouns [23, 3] or adjectives (attributes) [9, 26]. More recently, efforts are being made to predict entire sentences from image features [8, 25, 38, 19]. Some methods generate novel sentences by leveraging existing object detectors [12], attributes predictors [9, 4, 26], language statistics [38] or spatial relationships [19]. Sentences have also been assigned to images by selecting a complete written description from a large set [10, 25]. Our work focuses on the reverse problem of generating or retrieving images for textual descriptions. This leads to significantly different approaches, e.g. our model only requires pairwise potentials to model relative position, where [19] requires trinary potentials. Our potentials model a rich variety of visual information, including expression, pose and gaze. Of course at the core, our approach learns the visual meaning of semantically rich text that may also help produce more grounded textual descriptions of images.

Text to images (retrieval): Many efforts have been made in the computer vision and multimedia community to improve image search from textual queries. Some approaches build intermediate representations of images that capture mid-level semantic concepts [34, 30, 24, 40, 7, 35] and help bridge the well known semantic gap. These semantic concepts or attributes can also be used to pose queries for image search [20, 33]. Statements about relative attributes can be used to refine search results [18]. Our work allows for significantly richer textual descriptions (e.g. sets of complete sentences) as queries. In that sense, most related to ours is the work of Farhadi *et al*. [10]. Their "meaning" representation only involved tuples of the (object, action, scene) form. As will be evident later, we allow for a significantly larger variety of textual phrases and learn their visual meaning from data. We demonstrate this by *generating* novel scenes in addition to retrieving scenes as in [10].

Text to images (generation): In computer graphics the use of sentence descriptions has been used as an intuitive method for generating static scenes [6] and animations [27]. Joshi *et al*. [17] extract keywords from a story and attempt to find real pictures for illustration. Gobron *et al*. [14] propose an interesting application where they build an emotion analyzer from text. The emotions are then rendered using a human avatar.

Beyond nouns: Many papers have explored the visual meaning of different parts of speech beyond simple nouns. Attribute-based representations [22, 9, 26] typically detect adjectives. Many prepositions convey the spatial relations [5] between objects. Gupta *et al*. [16] explore the use of both prepositions and adjectives to build better object models, while [31] and [39] study relations that convey information related to active verbs, such as "riding" or "playing". Finally, our work follows [41] which studies the relationships between individual words and visual features using abstract scenes. We extend this work to learn the meaning of semantic phrases extracted from a sentence, which convey complex information related to numerous visual features.

3. Scene model

We begin by describing our model for scene generation using Conditional Random Fields (CRFs). Our approach to sentence parsing and how the parsed sentences are used to determine the CRF potentials is described in following sections. We use scenes that are created from 80 pieces of clip art representing 58 different objects [41], such as people, animals, toys, trees, etc. The dataset was created using Amazon's Mechanical Turk (AMT). Turkers were allowed

Jenny was mad and tried to kick Mike.
<<Jenny>, <be mad>, <>>
<<Jenny>, <try>, <kick>>
<<Jenny>, <kick>, <Mike>>

Mike and Jenny are wearing hats.
<<Mike>, <wear>, <hat>>
<< Jenny>, <wear>, <hat>>

Mike is holding a baseball bat.
<<Mike>, <hold>, <bat>>

Mike is standing in front of the table.
<<Mike>, <stand in_front_of>, <table>>

Jenny is sad because she wants the ball.
<<Jenny>, <be sad>, <>>
<<she>, <want>, <ball>>

Mike is wearing a blue hat with a star.
<<Mike>, <wear>, <hat>>
<<hat>, <with>, <star>>

Jenny is happy to see Mike.
<<Jenny>, <be happy>, <>>
<<Jenny>, <see>, <Mike>>

Mike is eating a burger
<<Mike>, <eat>, <burger>>

The cat is watching Jenny and Mike.
<<cat>, <watch>, <Jenny>>
<<cat>, <watch>, <Mike>>

Jenny wants Mike to share the bat.
<<Jenny>, <want>, <share>>
<<Mike>, <share>, <bat>>

Jenny is on the swings.
<<Jenny>, <be>, <>>

Figure 2: Example tuples extracted from sentences. Correct tuples are shown in blue, and incorrect or incomplete tuples are shown in red (darker).

to place objects anywhere in the scene. The Turkers could flip the clip art horizontally and choose between three discrete scales or depths when placing clip art. Example scenes are shown throughout the paper.

We model scenes using a fully connected CRF [21, 32] where each node represents an object, and edges represent their pairwise relations. The CRF allows us to compute the conditional probability of a scene given a set of sentences. An object is modeled using three types of parameters. The occurrence of object i is represented using a binary variable c_i. $\Phi_i = \{x_i, y_i, z_i, d_i\}$ models the 3D position and and the direction $d_i \in \{-1, 1\}$ the object is facing (left or right). Finally, if the object is a person it has a set of discrete attributes $\Psi_i = \{e_i, g_i, h_i\}$ that encode the person's expression e_i, pose g_i and clothing h_i. While we currently only model attributes for people, the model may handle attributes for other objects as well. We define the conditional probability of the scene $\{c, \Phi, \Psi\}$ given a set of sentences S as

$$\log P(c,\Phi,\Psi|S,\theta) = \sum_i \left(\overbrace{\psi_i(c_i,S;\theta_c)}^{occurrence} + \overbrace{\lambda_i(\Phi_i,S;\theta_\lambda)}^{abs.\ location} + \overbrace{\pi_i(\Psi_i,S;\theta_\pi)}^{attributes} \right) + \sum_{ij} \overbrace{\phi_{ij}(\Phi_i,\Phi_j,S;\theta_\phi)}^{rel.\ location} - \log Z(S,\theta) \tag{1}$$

where ψ, λ and π are the unary potentials, ϕ is the pairwise potential and $Z(S,\theta)$ is the partition function that normalizes the distribution. The variables i and j index the set of objects, and θ represents the model's parameters. We now describe how we compute each potential in turn.

Occurrence: We compute the unary occurrence potential using

$$\psi_i(c_i, S; \theta_c) = \log \theta_\psi(c_i, i, S) \tag{2}$$

where the parameters $\theta_\psi(c_i, i, S)$ encode the likelihood of observing or not observing object i given the sentences S. We describe how we compute θ_ψ in Section 5.

Absolute location: We compute an object's absolute location potential using a Gaussian Mixture Model (GMM),

$$\lambda_i(\Phi_i, S; \theta_\lambda) = \log \sum_k P(\Phi_i|k)\theta_\lambda(i,k), \tag{3}$$

where $P(\Phi_i|k) = \mathcal{N}((x_i, y_i); \mu_k(z_i), \sigma_k(z_i))$. Note that the mixture components are shared across all object types. Their parameters are learned using the K-means algorithm (9 components). Each object is assigned one of a discrete set of depths, z_i. A separate set of components are learned for each depth level. The model parameters $\theta_\lambda(i,k)$ are set equal to the empirical likelihood $P(k|i)$ of the kth component given the object i in the training data. Absolute location priors for some objects are shown in Figure 3.

Attributes: The attribute potential encodes the likelihood of observing the attributes given the sentences if the object is a person and is 0 otherwise

$$\pi_i(\Psi_i, S; \theta_\pi) = \left\{ \begin{array}{cc} \log \sum_k \theta_\pi(\Psi_{ik}, i, k, S) & person \\ 0 & otherwise \end{array} \right\} \tag{4}$$

The variable k indexes the set of binary attributes containing 5 expressions e_i, 7 poses g_i and 10 wearable items h_i (hats and glasses.) We discuss how we learn the parameters θ_π in Section 5.

Relative location: The pairwise potential Φ models the relative location of pairs of objects. We model the objects' relative 2D image position separately from the relative depth,

$$\phi_{ij}(\Phi_i, \Phi_j, S; \theta_\phi) = \log \sum_k \overbrace{P(\Delta_x, \Delta_y|k)\theta_{\phi,xy}(i,j,k,S)}^{relative\ 2D\ location} + \log \overbrace{\theta_{\phi,z}(i,j,\Delta_z,S)}^{relative\ depth} \tag{5}$$

The relative 2D location is modeled using a GMM similar to the absolute spatial location in Equation (3). The parameters $\theta_{\phi,xy}(i,j,k,S)$ compute the likelihood $P(k|i,j,S)$ of the kth component given the object types and sentences. We discuss how these parameters are computed in Section 5. The relative position (Δ_x, Δ_y) of object i to object j is computed as

$$\Delta_x = \left\{ \begin{array}{cc} x_i - x_j & d_i = -1 \\ x_j - x_i & d_i = 1 \end{array} \right\} \tag{6}$$

and $\Delta_y = y_i - y_j$. We include the direction d_i object i is facing when computing Δ_x so that we may determine whether object i is facing object j. This is important especially for humans and animals where the eye gaze direction

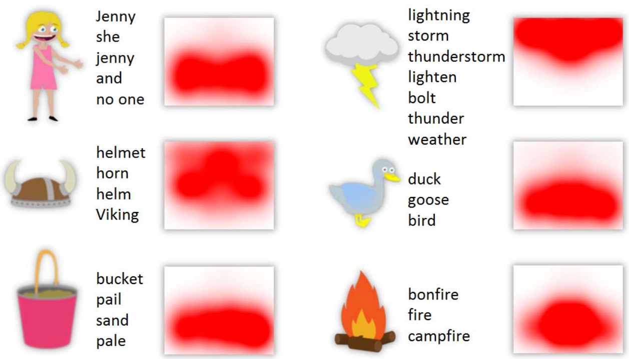


Figure 3: Examples of nouns with highest mutual information for six objects. The absolute spatial prior for each object is displayed in red. A list of the 200 most frequent nouns can be found in the supplementary material.

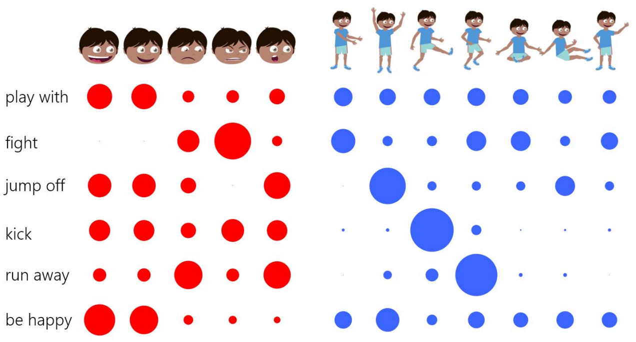


Figure 4: Figure showing the probability of expression (red) and pose (blue) for the primary object for several predicate relations, larger circle implies greater probability.

is semantically meaningful [41]. The value of $P(\Delta_x, \Delta_y|k)$ is computed using a standard normal distribution. Once again the means and standard deviations of the mixture components are shared among all object classes. This property is essential if we hope to learn relations that generalize across objects, i.e. that "next to" implies the same spatial relationship regardless of the objects that are "next to" each other. The parameters of the mixture components are learned using the K-means algorithm $(24$ components$)$. Note the values of the mixture components inherently encode co-occurrence information, i.e. their values are larger for objects that commonly co-occur.

The parameters $\theta_{\phi,z}(i, j, \Delta_z, S)$ used by the relative depth potential encode the probability of the depth ordering of two objects given the sentences $P(\Delta_z|i, j, S)$. Δ_z has three discrete values $\{-1, 0, 1\}$ corresponding to whether object i is behind object j, at the same depth, or in front of object j. We describe how we compute the parameters $\theta_{\phi,z}$ in Section 5.

4. Sentence parsing

In the previous section we described our CRF model for scene generation. A majority of the model's parameters for computing the unary and pairwise potentials are dependent on the set of given sentences S. In this section we describe how we parse a set of sentences into a set of predicate tuples. In the following section we describe how to determine the CRF parameters given the predicate tuples.

A set of predicate tuples is a common method for encoding the information contained in a sentence. Several papers have also explored the use of various forms of tuples for use in semantic scene understanding [8, 31, 19]. In our representation a tuple contains a primary object, a relation, and an optional secondary object. The primary and secondary object are both represented as nouns, where the relation may take on several forms. The relation may be a single word, such as a verb or preposition, or it may be a combination of multiple words such as <verb, preposition> or <verb, adjective> pairs. Examples of sentences and tuples are shown in Figure 2. Note that each sentence can produce multiple tuples. The tuples are found using a technique called semantic roles analysis [28] that allows for the unpacking of a sentence's semantic roles into a set of tuples. Note that the words in the sentences are represented using their lemma or "dictionary look-up form" so that different forms of the same word are mapped together. For instance "run", "ran", "runs" are all mapped to "run". Each sentence may contain multiple tuples. Finally, while we model numerous relationships within a sentence, there are many we do not model and semantic roles analysis often misses tuples and may contain errors. One notable relation not currently modeled by our system is attributive adjectives. For instance "happy" in "The happy boy" is a attributive adjective that we do not capture. However, "happy" in "The boy is happy" is a predicative adjective that is modeled by our tuple extractor. For semantic roles analysis we use the code supplied online by the authors of [28].

In our experiments we use a set of 10,000 clip art scenes provided by [41]. For each of these scenes we gathered two sets of descriptions, each containing three sentences using AMT. The turkers were instructed to "Please write three simple sentences describing different parts of the scene. These sentences should range from 4 to 8 words in length using basic words that would appear in a book for young children ages 4-6." 9,000 scenes and their 54,000 sentences were used for training, and 1000 descriptions (3000 sentences) for the remaining 1000 scenes were used for testing. 319 nouns (objects) and 445 relations were found at least 8 times in the sentences.

5. Scene generation

In this section, we assume each scene has a set of tuples $T = \{t_1, \dots, t_n\}$. Each tuple t_i contains three indices $\{p_i, r_i, s_i\}$ corresponding to the tuple's primary object $p_i \in Q$, relation $r_i \in R$ and secondary object $s_i \in Q$, where Q is the set of objects and R the set of relations. We now describe how the tuples are used to compute the CRF's parameters for scene generation, followed by how we generate scenes using the CRF.

Each tuple contains one or two nouns and a relation that

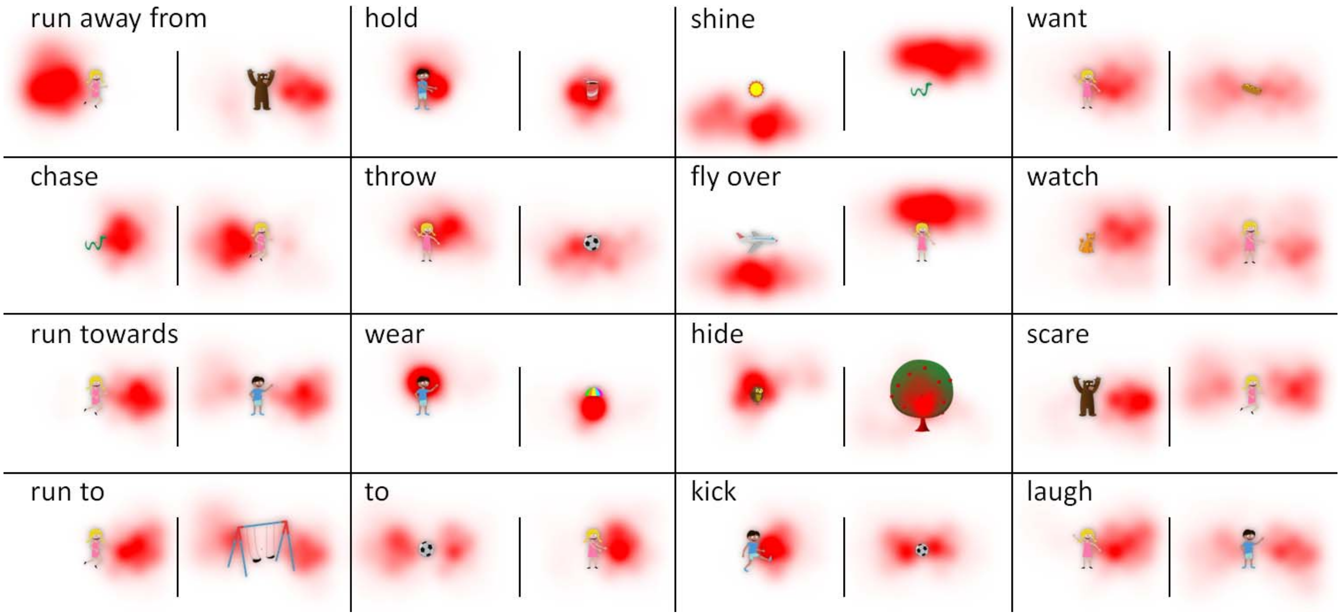


Figure 5: Illustration of the relative spatial locations of the primary (left) and secondary (right) objects for 16 relations. The relative location distributions are shown in red. The most likely primary and secondary object for each relation is shown for context. Notice how some relations convey very different relative spatial locations, such as "run away from" and "chase". Other relations, such as "want", "watch", "scare", and "laugh" all share very similar spatial relations. A list of the 300 most frequently found relations can be found in the supplementary material.

may relate to the objects' relative positions, attributes, etc. We assume the nouns only provide information related to the occurrence of objects. We make the simplifying assumption that each noun used by the primary or secondary object only refers to a single object in the scene. Our first task is to assign one of the 58 unique objects in the clip art collection to each noun. We perform this mapping by finding the clip art object that has the highest mutual information for each noun over all of the training scenes $\mathcal{M}(i) = \max_j I(i;j)$, where $I(i;j)$ is the mutual information between noun i and clip art object j. The nouns with highest mutual information are shown for several objects in Figure 3. We found this approach to be surprisingly effective in finding the correct mapping in nearly all instances. The main failures were for ambiguous nouns such as "it" and nouns for which no distinct clip art exists, such as "ground", "sky", or "hand".

The noun mapping $\mathcal{M}$ may be used to map both the primary p_i and secondary s_i objects to specific pieces of clip art in the scene. This information may be used to update the parameters $\theta_\psi(c_i, i, S)$ of the unary occurrence potentials in the CRF. Intuitively, we compute the potentials such that all objects in the tuples are contained in the scene and otherwise their occurrence is based on their prior probability. If $j = \mathcal{M}(p_i)$ we update $\theta_\psi(c_j, j, S) = 1$ if the primary object is present $c_j = 1$ and 0 otherwise. We perform a similar update for the secondary object if it exists. For objects j not included in a tuple, $\theta_\psi(c_j, j, S)$ is set to their prior probability of occurring in a scene.

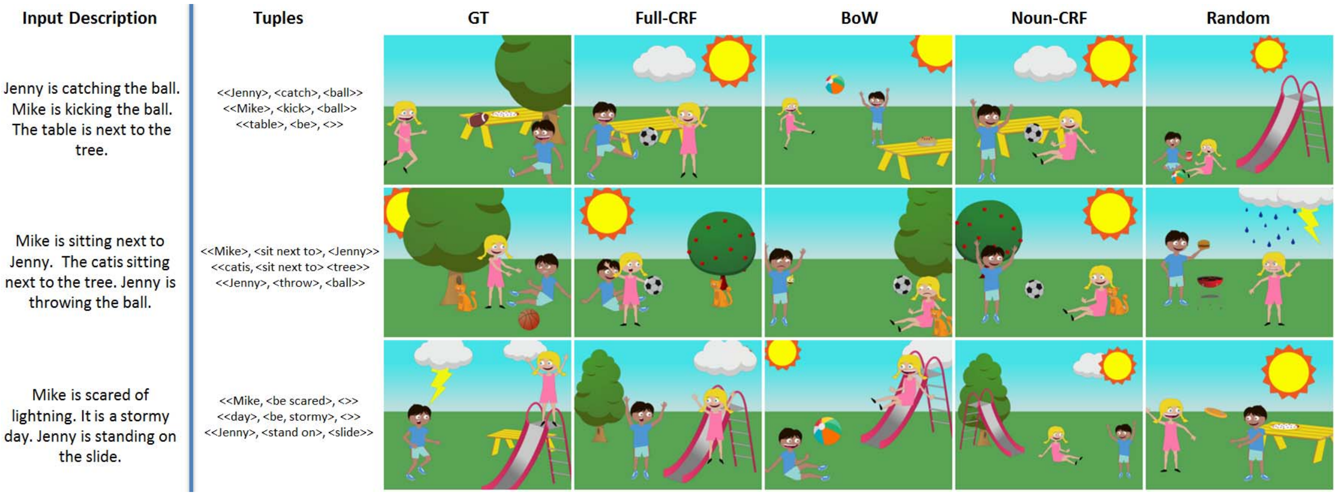
Next, we consider how the relation $r_i \in R$ is used to update the other CRF parameters. For each relation $l \in R$, we empirically compute two sets of values corresponding to the likelihood of the primary object's attributes $P_p(k|l)$ and secondary object's attributes $P_s(k|l)$ given the relation l, where k is an index over the set of attributes. For instance we compute the empirical probability of the "smile" expression attribute for the primary object given the "laughing at" relation. Using $P_p(k|r_i)$ and $j = \mathcal{M}(p_i)$ we update the attribute parameters for each object j and attribute k using

$$\theta_\pi(\Psi_{jk}, j, k, S) = \left\{ \begin{array}{cc} P_p(k|r_i) & \Psi_{jk} = 1 \\ 1 - P_p(k|r_i) & \Psi_{jk} = 0 \end{array} \right\}, \quad (7)$$

and similarly for the secondary object if it exists. Example attribute empirical probabilities for several relations are shown in Figure 4. If an object is a member of multiple tuples, the average values of $P_p(k|r_i)$ or $P_s(k|r_i)$ across all tuples are used. For all objects not members of a tuple, the attribute parameters are set equal to the empirical prior probability of the attributes given the object.

Given the mappings $\mathcal{M}(p_i)$ and $\mathcal{M}(s_i)$, we know the location of the primary and secondary object for all instances of the relation $l \in R$ in the training data. Thus, we compute the empirical likelihood $P(k|l)$ of the kth component of the GMM used to model the relative 2D location of $\mathcal{M}(p_i)$ and $\mathcal{M}(s_i)$ given relation $l \in R$. Similarly, we compute the empirical likelihood $P(\Delta_z|l)$ of the relative depth ordering between objects. Using $P(k|r_i)$, $j = \mathcal{M}(p_i)$, and $j' = \mathcal{M}(s_i)$ we set the relative position parameters of the CRF using

$$\theta_{\phi,xy}(j, j', k, S) = P(k|r_i), \quad (8)$$



Jenny is catching the ball. Mike is kicking the ball. The table is next to the tree.

<<Jenny>, <catch>, <ball>>
<<Mike>, <kick>, <ball>>
<<table>, <be>, <>>

Mike is sitting next to Jenny. The catis sitting next to the tree. Jenny is throwing the ball.

<<Mike>, <sit next to>, <Jenny>>
<<catis, <sit next to> <tree>>
<<Jenny>, <throw>, <ball>>

Mike is scared of lightning. It is a stormy day. Jenny is standing on the slide.

<<Mike, <be scared>, <>>
<<day>, <be, stormy>, <>>
<<Jenny>, <stand on>, <slide>>

Figure 6: Three random example scenes generated by our approach (Full-CRF) for the given input description (left). The resultant tuples are also shown (2^{nd} column). Our scenes faithfully reproduce the meaning of the input descriptions. In the first example (top), our scene has a soccer ball instead of the foot ball in the ground truth scene (GT). This is because the input description did not specify the type of ball. The tree mentioned in the description is missing in our scene because the tuples missed the tree. The bag-of-words model (BoW) can not infer who is kicking the ball, and hence returns an image where Jenny is kicking the ball instead of Mike. The CRF model updated with only the nouns (Noun-CRF) contains the right objects, but at locations and relationships inconsistent with the input description. Random scenes from the dataset (Random) are also shown to demonstrate the variety of scenes in the dataset. More examples can be found in the supplementary material.

and the depth parameters by

$$\theta_{\phi,z}(j, j', \Delta_z, S) = P(\Delta_z | r_i). \tag{9}$$

Note the values of $P(k|r_i)$ and $P(\Delta_z|r_i)$ are not dependent on the object types. Thus we may learn the generic properties of a relation such as "next to" that is not dependent on the specific pair of objects that are next to each other. This allows us to generalize the relations to object pairs that may not have been present in the training data. Examples of the relative spatial locations learned for different relations are shown in Figure 5. Notice the variety of spatial relations captured. Interestingly, "want", "watch", "scare" and "laugh" all learn similar relative spatial locations. Intuitively, this makes sense since each relation implies the primary object is looking at the secondary object, but the secondary object may be facing either direction. If the pairwise parameters are not specified by the relations in the tuples, the parameters are set to the empirical prior probabilities given the object types. If a tuple does not contain a secondary object, the tuple is not used to update the parameters for the pairwise potentials.

5.1. Generating scenes using the CRF

After the potentials of the CRF have been computed given the tuples extracted from the sentences, we generate a scene using a combination of sampling and iterated conditional modes. Since there are 58 objects and each object has a 3D position, an orientation and possible attributes, a purely random sampling approach would require a prohibitively large number of random samples to find a high likelihood scene. Instead we iteratively select at random a single object i and determine $\{c_i, \Phi_i, \Psi_i\}$ assuming the other objects' assignments are fixed. The occurrence of the selected object i is randomly sampled with probability $\theta_\psi(1, i, S)$. If it is visible we apply an approach similar to iterated conditional modes that maximizes the joint probability of the CRF. That is, the most probable position given the location of the other objects is chosen. Similarly, the most likely orientation d_i for the object is chosen. If the object is a person, the most likely expression and pose are chosen. If the object is something that may be worn such as a hat or glasses, its position is determined by the person whose attributes indicate it is most likely to be worn by them. If a person is not present, the worn object's position is determined similarly to other objects. This describes our approach to generating one sample. We generate 30,000 such samples and pick the one with the maximum probability. Examples of generated scenes are shown in Figure 6. The qualitative results shown in Figures 3 to 6 show the algorithm is learning intuitively "correct" interpretations of semantic phrases that will likely transfer to real images. Evaluating these models on real images is part of future work.

6. Results

We evaluate our approach on two tasks: scene generation and image retrieval.

6.1. Scene Generation

We use each one of our 1000 test descriptions (each description is a set of 3 sentences) as input, and generate a scene using our approach. We conduct human studies to evaluate how well our generated scene matches the input

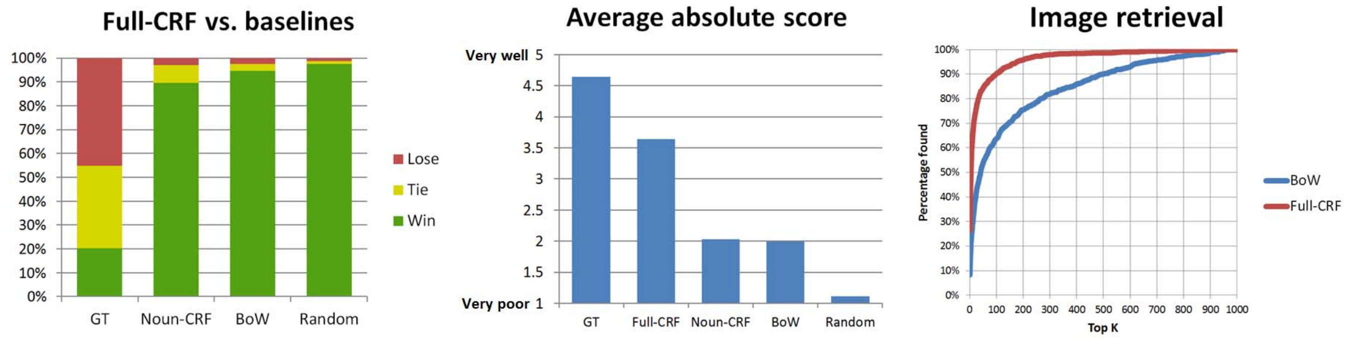


Figure 7: (left) The results of a human study asking which scenes better depicted a set of sentences. The subjects find our scenes (Full-CRF) better represent the input sentences than all baseline approaches. In fact, our approach wins over or ties with the ground truth scenes frequently. (middle) Subjects were asked to score how well a scene depicted a set of three sentences from 1 (very poor) to 5 (very well). We achieve absolute scores slightly worse than the ground truth, but better than the baselines. (right) Our approach finds more correct matches in the top K retrieved images than a bag-of-words baseline.

description. We compare our approach (Full-CRF) to the following baselines. **GT**: The ground truth uses the original scenes that the mechanical turkers' viewed while writing their sentence descriptions. Since all of these scenes should provide good matches to the sentences, the best we can expect from our approach is to tie with the ground truth. **BoW:** We build a bag-of-words representation for the input description that captures whether a word (primary object, secondary object or relation) is present in the description or not. [1] Using this representation, we find the most similar description from the training dataset of $9,000$ scenes. The corresponding scene is returned as the output scene. The same NLP parsing was used for this baseline as our approach. Notice that this baseline does not *generate* a novel scene. **Noun-CRF:** This baseline generates a scene using the CRF, but only based on the primary and secondary object nouns present in the predicate tuples. The tuple's relation information is not used, and the corresponding potentials in the CRF use the training dataset priors. **Random:** We pick a random scene from the training data.

We conducted our user studies on Amazon Mechanical Turk. We paired our result with each of the above 4 baselines for all 1000 test descriptions. Subjects were shown the input description and asked which one of the two scenes matched the description better, or if both equally matched. Five subjects were shown each pair of scenes. The results are shown in Figure 7 (left). We also conducted a study where subjects were shown the input description and the output scene and asked on a scale of 1 (very poorly) - 5 (very well), how well the scene matched the description. Results are shown in Figure 7 (middle). We see that our approach significantly outperforms all baselines on both tasks. It is especially notable that our approach wins over or ties with the ground truth scenes (GT) in 50% of the examples. In terms of the absolute scores, our approach scores a respectable average of 3.46 compared to the score of 4.64 for the ground truth scenes. The fact that our approach significantly outperforms the bag-of-words nearest neighbor baseline (BoW) (1.99) and the nouns-only CRF (Noun-CRF) baseline (2.03) shows that it is essential to learn the semantic meaning of complex language structures that encode the relationships among objects in the scene. As expected the random baseline performs the worst (1.11), but it demonstrates that the dataset is challenging in that random scenes rarely convey the same semantic meaning. Some randomly chosen qualitative results are shown in Figure 6, and additional results may be viewed in the supplementary material.

6.2. Image Retrieval

Given an input test description (i.e. a user-provided query), we use our CRF to score all 1000 test scenes in the dataset. We sort the images by this score and return the top K images. We report the percentage of queries that return the true target image in the top K images. We compare results for varying values of K to the BoW baseline that only matches tuple objects and relations extracted from a separate set of 3,000 training sentences on the 1000 test scenes. The BoW baseline does not use any visual features. Results are shown in Figure 7 (right). We see that our approach significantly outperforms the baseline. A user would have to browse through only a tenth of the images using our approach as compared to the baseline to achieve the same recall of 75%. On average, our approach ranks the true target image at 37 compared to 150 by the baseline. This helps demonstrate that obtaining a deeper visual interpretation of a sentence significantly improves the quality of descriptive text-based queries.

7. Discussion

The unreliability of current detectors on real images has limited our ability to take a step forward and research complex semantic relations to visual data. Hence, many papers [2, 16, 17, 19] only learn a relatively small number of relations (19 in [16]). Our paper is the first to reason about $>$ 400 diverse relations (combinations of verbs, adjectives, prepositions) containing subtle differences between concepts such as "ran after" and "ran to." Furthermore, pre-

[1] Entire tuples occur too rarely to be used as "words".

vious works learn relations using only occurrence [2, 16] and relative position [16]. As we demonstrate, complex semantic phrases are dependent on a large variety of visual features including object occurrence, relative position, facial expression, pose, gaze, etc. This paper only begins to explore the numerous challenges and interesting problems in semantic scene understanding.

One critical aspect of our approach is the extraction of predicate tuples from sentences. Currently, many tuples are missed or incorrect. This may be due to the failure of the semantic roles analysis algorithm or to grammatically incorrect sentences. In either case, improving tuple extraction is an important area for future research. In fact it may be beneficial to extract too many tuples, and let the scene model determine which are correct. For instance, sentences with ambiguous phrase attachment, such as ”Jenny ran after the bear with a bat.” may be correctly interpreted, *i.e*. Jenny has the bat, not the bear.

While we study the problem of scene generation and retrieval in this paper, the features we learn may also be useful for generating rich and descriptive sentences from scenes. This exciting area of future research could be carried out using the same dataset described in this paper.

In conclusion we demonstrate a method for automatically inferring the visual meaning of predicate tuples extracted from sentences. The tuples relate one or two nouns using a combination of verbs, prepositions and adjectives. We show our model is capable of generating or retrieving scenes that correctly interpret sets of sentences.

References

[1] A. Aker and R. Gaizauskas. Generating image descriptions using dependency relational patterns. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 1250–1258. Association for Computational Linguistics, 2010.

[2] K. Barnard, P. Duygulu, D. Forsyth, N. De Freitas, D. M. Blei, and M. I. Jordan. Matching words and pictures. *The Journal of Machine Learning Research*, 3:1107–1135, 2003.

[3] K. Barnard and Q. Fan. Reducing correspondence ambiguity in loosely labeled training data. In *CVPR*, 2007.

[4] T. Berg, A. Berg, and J. Shih. Automatic attribute discovery and characterization from noisy web data. *ECCV*, 2010.

[5] I. Biederman, R. Mezzanotte, and J. Rabinowitz. Scene perception: Detecting and judging objects undergoing relational violations. *Cognitive psychology*, 14(2), 1982.

[6] B. Coyne and R. Sproat. Wordseye: an automatic text-to-scene conversion system. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 487–496. ACM, 2001.

[7] M. Douze, A. Ramisa, and C. Schmid. Combining attributes and fisher vectors for efficient image retrieval. In *CVPR*, 2011.

[8] A. Farhadi, I. Endres, and D. Hoiem. Attribute-centric recognition for cross-category generalization. In *CVPR*, 2010.

[9] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, 2009.

[10] A. Farhadi, M. Hejrati, A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth. Every picture tells a story: Generating sentences for images. In *ECCV*, 2010.

[11] B. Fasel and J. Luettin. Automatic facial expression analysis: a survey. *Pattern Recognition*, 36(1), 2003.

[12] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 32(9), 2010.

[13] Y. Feng and M. Lapata. How many words is a picture worth? automatic caption generation for news images. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1239–1249. Association for Computational Linguistics, 2010.

[14] S. Gobron, J. Ahn, G. Paltoglou, M. Thelwall, and D. Thalmann. From sentence to emotion: a real-time three-dimensional graphics metaphor of emotions extracted from text. *The Visual Computer*, 26(6-8):505–519, 2010.

[15] M. Grubinger, P. Clough, H. Müller, and T. Deselaers. The iapr tc-12 benchmark: A new evaluation resource for visual information systems. In *Int. Workshop OntoImage*, 2006.

[16] A. Gupta and L. Davis. Beyond nouns: Exploiting prepositions and comparative adjectives for learning visual classifiers. *ECCV*, 2008.

[17] D. Joshi, J. Z. Wang, and J. Li. The story picturing engine—a system for automatic text illustration. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, 2(1):68–89, 2006.

[18] A. Kovashka, D. Parikh, and K. Grauman. Whittlesearch: Image search with relative attribute feedback. In *CVPR*, 2012.

[19] G. Kulkarni, V. Premraj, S. Dhar, S. Li, Y. Choi, A. Berg, and T. Berg. Baby talk: Understanding and generating simple image descriptions. In *CVPR*, 2011.

[20] N. Kumar, P. Belhumeur, and S. Nayar. Facetracer: A search engine for large collections of images with faces. In *ECCV*, 2010.

[21] J. Lafferty, A. McCallum, and F. C. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001.

[22] C. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009.

[23] V. Lavrenko, R. Manmatha, and J. Jeon. A model for learning the semantics of pictures. In *NIPS*, 2003.

[24] M. Naphade, J. Smith, J. Tesic, S. Chang, W. Hsu, L. Kennedy, A. Hauptmann, and J. Curtis. Large-scale concept ontology for multimedia. *IEEE Multimedia*, 2006.

[25] V. Ordonez, G. Kulkarni, and T. Berg. Im2text: Describing images using 1 million captioned photographs. In *NIPS*, 2011.

[26] D. Parikh and K. Grauman. Relative attributes. In *ICCV*, 2011.

[27] K. Perlin and A. Goldberg. Improv: A system for scripting interactive actors in virtual worlds. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 205–216. ACM, 1996.

[28] C. Quirk, P. Choudhury, J. Gao, H. Suzuki, K. Toutanova, M. Gamon, W.-t. Yih, L. Vanderwende, and C. Cherry. Msr splat, a language analysis toolkit. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Demonstration Session*, pages 21–24. Association for Computational Linguistics, 2012.

[29] C. Rashtchian, P. Young, M. Hodosh, and J. Hockenmaier. Collecting image annotations using amazon's mechanical turk. In *NAACL HLT Workshop on Creating Speech and Language Data with Amazon's MT*, 2010.

[30] N. Rasiwasia, P. Moreno, and N. Vasconcelos. Bridging the gap: Query by semantic example. *IEEE Transactions on Multimedia*, 2007.

[31] M. Sadeghi and A. Farhadi. Recognition using visual phrases. In *CVPR*, 2011.

[32] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *International Journal of Computer Vision*, 81(1):2–23, 2009.

[33] B. Siddiquie and A. Gupta. Beyond active noun tagging: Modeling contextual interactions for multi-class active learning. In *CVPR*, 2010.

[34] J. Smith, M. Naphade, and A. Natsev. Multimedia semantic indexing using model vectors. In *ICME*, 2003.

[35] X. Wang, K. Liu, and X. Tang. Query-specific visual semantic spaces for web image re-ranking. In *CVPR*, 2011.

[36] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*. IEEE, 2010.

[37] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR*, 2011.

[38] Y. Yang, C. Teo, H. Daumé III, and Y. Aloimonos. Corpus-guided sentence generation of natural images. In *EMNLP*, 2011.

[39] B. Yao and L. Fei-Fei. Modeling mutual context of object and human pose in human-object interaction activities. In *CVPR*, 2010.

[40] E. Zavesky and S.-F. Chang. Cuzero: Embracing the frontier of interactive visual search for informed users. In *Proceedings of ACM Multimedia Information Retrieval*, 2008.

[41] C. Zitnick and D. Parikh. Bringing semantics into focus using visual abstraction. In *CVPR*, 2013.