Learning Common Sense Through Visual Abstraction

Ramakrishna Vedantam[1*] Xiao Lin[1*] Tanmay Batra[2†] C. Lawrence Zitnick[3] Devi Parikh[1]
[1]Virginia Tech [2]Carnegie Mellon University [3]Microsoft Research
[1]{vrama91,linxiao,parikh}@vt.edu [2]tbatra@andrew.cmu.edu [3]larryz@microsoft.com

Abstract

Common sense is essential for building intelligent machines. While some commonsense knowledge is explicitly stated in human-generated text and can be learnt by mining the web, much of it is unwritten. It is often unnecessary and even unnatural to write about commonsense facts. While unwritten, this commonsense knowledge is not unseen! The visual world around us is full of structure modeled by commonsense knowledge. Can machines learn common sense simply by observing our visual world? Unfortunately, this requires automatic and accurate detection of objects, their attributes, poses, and interactions between objects, which remain challenging problems. Our key insight is that while visual common sense is depicted in visual content, it is the semantic features that are relevant and not low-level pixel information. In other words, photorealism is not necessary to learn common sense. We explore the use of human-generated abstract scenes made from clipart for learning common sense. In particular, we reason about the plausibility of an interaction or relation between a pair of nouns by measuring the similarity of the relation and nouns with other relations and nouns we have seen in abstract scenes. We show that the commonsense knowledge we learn is complementary to what can be learnt from sources of text.

1. Introduction

Teaching machines common sense has been a longstanding challenge at the core of Artificial Intelligence (AI) [8]. Consider the task of assessing how plausible it is for a dog to jump over a tree. One approach is to mine text sources to estimate how frequently the concept of dogs jumping over trees is mentioned. A long history of works address the problem is this manner by mining knowledge from the web [5, 21, 24] or by having humans manually specify facts [4, 28, 33, 34] in text. Unfortunately, text is known to suffer from a reporting bias. If the frequency of mention was an indication of occurrence in the real world, people are ∼3 times more likely to be murdered than they are to inhale, and people inhale ∼6 times as often as they exhale [16]. This bias is not surprising. After all, people talk about things that are interesting to talk about, and unusual circumstances tend to be more interesting.

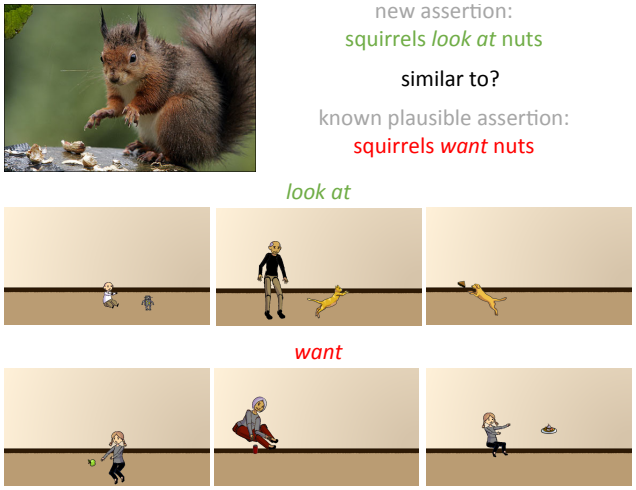


Figure 1: We consider the task of assessing how plausible a commonsense assertion is based on how similar it is to known plausible assertions. We argue that this similarity should be computed not just based on the text in the assertion, but also based on the visual grounding of the assertion. While "wants" and "looks at" are semantically different, their visual groundings tend to be similar. We use abstract scenes made from clipart to provide the visual grounding.

While unwritten, commonsense knowledge is not unseen! The visual world around us is full of structure modeled by our commonsense knowledge. By reasoning visually about a concept we may be able to estimate its plausibility more accurately. For instance, while "squirrels wanting nuts" is frequently mentioned in text, "squirrels looking at nuts" is rarely mentioned even though it is equally plausible. However, if we visually imagine a squirrel wanting a nut, we typically imagine a squirrel looking at a nut (Figure 1). This is because wanting something and looking at something tend to be visually correlated, even though

*Equal contribution
†This work was done as an intern at Virginia Tech

they have differing underlying meaning. Interestingly, in the word2vec [27] text embedding space that is commonly used to measure word similarity, *look at* is more similar to *feel* than to *want*. Clearly, vision and text provide complementary signals for learning common sense.

Unfortunately, extracting commonsense knowledge from visual content requires automatic and accurate detection of objects, their attributes, poses, and interactions. These remain challenging problems in computer vision. Our key insight is that commonsense knowledge may be gathered from a high-level semantic understanding of a visual scene, and that low-level pixel information is typically unnecessary. In other words, photorealism is not necessary to learn common sense. In this work, we explore the use of human-generated abstract scenes made from clipart for learning common sense. Note that abstract scenes are inherently *fully* annotated, allowing us to exploit the structure in the visual world, while bypassing the difficult intermediate problem of training visual detectors.

Specifically, we consider the task of assessing the plausibility of an interaction or relation between a pair of nouns, as represented by a tuple (primary noun, relation, secondary noun) e.g., (boy, kicks, ball). As training data, we collect a dataset of tuples and their abstract visual illustrations made from clipart. These illustrations are created by subjects on Amazon Mechanical Turk (AMT). We use this to learn a scoring function that can score how well an abstract visual illustration matches a test tuple.

Given a previously unseen tuple, we assess its plausibility using both visual and textual information. A tuple is deemed plausible if it has high alignment with the training tuples and visual abstractions. When measuring textual similarity between tuples we exploit the significant progress that has been made in learning word similarities from web scale data using neural network embeddings [27, 29]. A tuple's alignment with the visual abstractions provides information on its visual plausibility. We model a large number of free form relations (213) and nouns (2466), which may form over $\approx$1 billion possible tuples. We show that by jointly reasoning about text and vision, we can assess the plausibility of commonsense assertions more accurately than by reasoning about text alone.

The rest of this paper is organized as follows. We discuss related work in Section 2. Our data collection methodology is described in Section 3. Our model for classifying novel commonsense assertions (tuples) as plausible or not is presented in Section 4. Section 5 describes our experimental setup, followed by quantitative and qualitative results in Section 6, and a conclusion in Section 7.

2. Related Work

Common sense and text. There is a rich line of works which learn relations between entities to build knowledge bases either using machine reading (e.g., Knowledge Vault [11], NELL [5], ReVerb [12]) or using collaboration within a community of users (e.g., Freebase [4], Wikipedia[1]). We make use of the ReVerb Information Extraction system to create our dataset of tuples (more details in Section 3.2). Our goal is to learn common sense from a complementary source: our visual world. A task closely related to learning common sense from text is answering questions. Systems such as IBM Watson [13] combine multiple text-based knowledge bases to answer factual questions. Our work focuses on combining different modalities of information (abstract scenes and text) for the task of assessing the plausibility of commonsense assertions.

Common sense and vision. A popular use of commonsense knowledge in vision has been for modeling context for improved recognition [10, 14, 17, 18, 20]. Recently, there has been a surge in interest in high-level "beyond recognition" tasks which can benefit from external knowledge beyond what is depicted in the image [3, 19, 23, 30, 31]. Zhu *et al*. [35] use attribute and action classification along with information from various textual knowledge bases to perform tasks like zero-shot affordance prediction for human-object interactions. Their dictionary of relations was specified manually and limited to 19 inter-object relations. We explore a larger number of *free-form* relations (213 in total) extracted from text. Johnson *et al*. [22] build a scene graph representation for image retrieval which models attribute and object relations. LEVAN [9] trains detectors for a variety of bigrams (e.g., jumping horse) from google n-grams using web-scale image data. NEIL [7] analyzes images on the web to learn visual models of objects, scenes, attributes, part-of, and other ontology relationships. Our focus is less on appearance models and more on the underlying semantics. Recent work has also looked at mining *semantic* affordances, *i.e*. inferring whether a given action can be performed on an object [6]. In contrast, we are interested in the more general problem of predicting the plausibility of interactions or relations between pairs of objects. Lin and Parikh [26] propose to learn visual common sense and use it to answer textual fill-in-the-blank and visual paraphrasing questions, by imagining a scene behind the text. While they model visual common sense in the context of a scene, our task is at a more atomic level – reasoning about the plausibility of a specific relation or interaction between pairs of objects. Most similar to ours is a concurrent work VisKE [32] which also studies the task of evaluating the plausibility of commonsense assertions using visual cues. Their visual cues are derived from webly-supervised detection models, while we use abstract scenes and text embeddings. A new test tuple can be processed almost instantaneously using our approach, while training their webly-supervised detector takes $\sim$30 minutes per tuple. It is con-

[1] http://www.wikipedia.org/

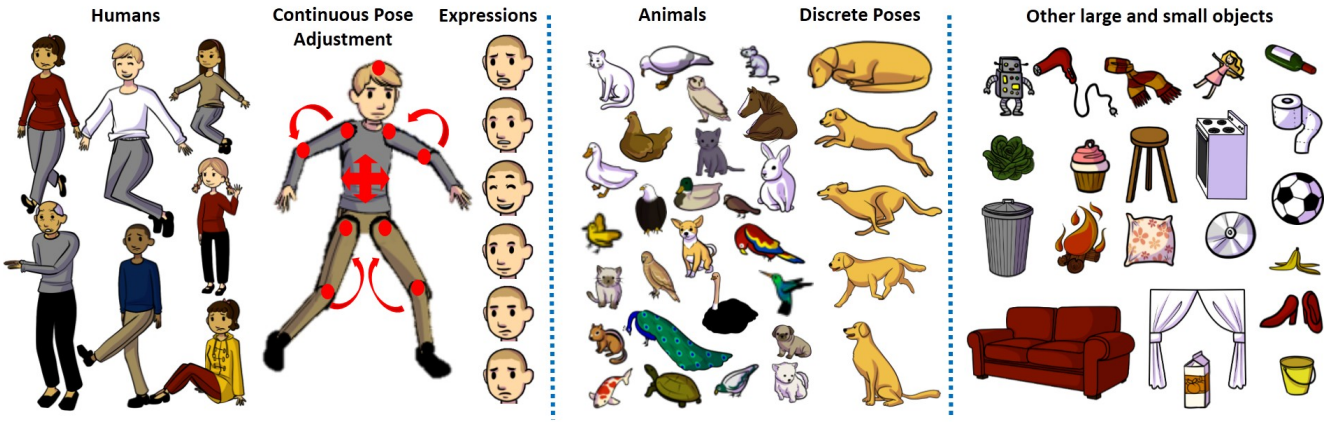


Figure 2: A subset of objects from our clipart library.

ceivable that text, abstract scenes and real images are all complementary sources of information.

Learning from visual abstraction. Visual abstractions have been explored for a variety of high-level scene understanding tasks. Zitnick and Parikh [37] learn the importance of various visual features (occurrence, co-occurrence, expression, gaze, *etc*.) in determining the meaning or semantics of a scene. Zitnick *et al*. also link the semantics of a scene to memorability and saliency of objects [36]. [38] learns the visual interpretation of sentences and generates scenes for a given input sentence. Fouhey and Zitnick [15] learn the dynamics of objects in scenes from temporal sequences of abstract scenes. Antol *et al*. [2] learn models of fine-grained interactions between pairs of people using visual abstractions, and evaluate their models on real images from the web. Lin and Parikh [26] "imagine" abstract scenes corresponding to text, and use the common sense depicted in these imagined scenes to solve textual tasks such as fill-in-the-blanks and paraphrasing. In this work, we are interested in using abstract scenes as a complementary source of commonsense knowledge to text for the task of classifying commonsense assertions as plausible or not.

3. Datasets

3.1. Abstract Scenes Vocabulary

In order to learn comprehensive commonsense knowledge, it is important for the library of clipart pieces to be expressive enough to model a wide variety of scenarios. Previous works on using visual abstractions depicted a boy and a girl playing in a park [15, 37, 38] with a library of 58 objects, or fine-grained interactions between two people [2] (no additional objects). Instead, our clipart library allows us to depict a variety of indoor scenes. It contains 20 "paper-doll" human models [2] spanning genders, races, and ages with 8 different expressions. The limbs are adjustable to allow for continuous pose variations. The vocabulary contains over 100 small and large objects and 31 animals in various poses, that can be placed at one of 5 discrete scales or depths in the scene, facing left or right. Our clipart is also more realistic looking than previous work. A snapshot of the library can be viewed in Figure 2. Note that while we restrict ourselves to indoor scenes in this work, our idea is general and applicable to other scenes as well. More clipart objects and scenes can be easily added to the clipart library.

3.2. Tuple Extraction

Extracting Seed Assertions: To collect a dataset of commonsense assertions, we start by extracting a set of seed tuples from image captions. We use the MS COCO training set [25] containing images annotated with 80 object categories and five captions per image. We pick a subset of 9913 images whose annotated objects all come from a list of manually selected objects from our library of clipart.[2] Note that MS COCO images are not fully annotated and contain many more objects than those annotated. As a result, captions for these images could contain nouns that may not be part of the annotated object list or our clipart library. Our model can handle this by using word embeddings as described in Section 4.1.

We split the images into VAL (4956 images) and TEST (4957 images). We then run the ReVerb [12] information extraction tool on the captions for these images (images are not involved anymore), along with some post-processing (described in supplementary) to obtain a set of (t_P, t_R, t_S) tuples, where t_P is the primary noun, t_R is the relation, and t_S is the secondary noun in the tuple t e.g., (plate, topped with, meat). All tuples containing relations that occur less than four times in the dataset are likely to be noisy extractions, and are removed. This gives us a set of 4848 tuples in

[2]List: *person*, *cat*, *dog*, *frisbee*, *bottle*, *wine glass*, *cup*, *fork*, *knife*, *spoon*, *apple*, *sandwich*, *hotdog*, *pizza*, *cake*, *chair*, *couch*, *potted plant*, *bed*, *dining table*, *tv*, *book*, *scissors*, *teddy bear* was selected to capture objects in our clipart library that are commonly found in living room scenes.

VAL and 4778 in TEST, 213 unique relations in VAL and 204 in TEST, and 2466 unique nouns in VAL and 2378 in TEST. VAL and TEST have 893 tuples, 814 nouns, and 151 relations in common. These tuples form our seed commonsense assertions.

Expanding Seed Assertions: We expand our seed set of assertions by generating random assertions. This is done on both TEST and VAL independently. We iterate through each tuple twice, and pair the corresponding t_R with a random t_P and t_S from all nouns that occur at least 10 times[3]. So there are twice as many expanded tuples as there are seed tuples. This results in 9700 expanded tuples in VAL and 9554 in TEST. Note that we are sampling from a space of 160 primary nouns (>10 occurrences) $\times$ 204 relations $\times$ 160 nouns i.e., >5 *million* possible TEST assertions. In total across seed and expanded, our VAL set contains 14548 commonsense assertions spanning 213 relations, and our TEST set contains $14,332$ commonsense assertions spanning 204 relations. To the best of our knowledge, ours is the first work that models such a large number of relations and commonsense assertions.

Supervision on Expanded Assertions: We then show our set of assertions (seed + expanded) to subjects on Amazon Mechanical Turk (AMT). We asked them to indicate if the scenario described by the assertion is typical or not. They are also given an option to flag scenarios that make no sense. We collect 10 judgments per assertion. A snapshot of this interface can be found in the supplementary material.

80.1% of annotations on seed tuples were positive. This is not surprising because these tuples were extracted from descriptions of images, and were thus clearly plausible. The creation of random expanded tuples predominantly adds negatives. But we found that some randomly generated assertions such as (puppy, lay next to, chair) and (dogs, lay next to, pepperoni pizza) were rated as plausible (positives). 15.3% of annotations on our expanded tuples were positive. Overall, 36% of the labels in VAL and 37% of the labels in TEST are positives.

3.3. Tuple Illustration Interface

We collect abstract illustrations for all 213 relations in VAL. We get each relation illustrated by 20 different workers on AMT using the interface shown in Figure 3. Each worker is shown a *background* scene and asked to modify it to contain the relation of interest. We used living room scenes from [1] as background scenes, which were realistic scenes created by AMT workers using the same abstract scenes vocabulary as ours (Section 3.1). Priming workers with different background scenes helps increase the diversity in the visual illustrations of relations. For instance, when asked to create a scene depicting 'holding', a majority of workers might default to thinking of a person holding something while standing. But if they are primed with a scene where a woman is already sitting on a couch, then they might place a glass in her hand to make her hold the glass, resulting in a sitting person holding something. Workers are then instructed to indicate which clipart pieces in the scene correspond to the primary and secondary objects participating in the relation, and name them using as few words as possible.

[3]This is a coarse proxy for sampling nouns proportional to how often they occur in the seed set.

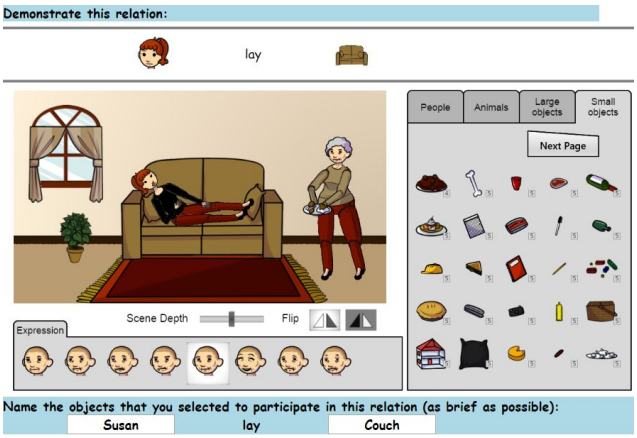


Figure 3: Our tuple illustration AMT interface.

To summarize, we collect 20 scenes depicting each of the 213 relations in VAL (4260 scenes total), along with annotations for the primary and secondary nouns and corresponding clipart objects participating in the relation. These form our set of TRAIN tuples that will be used to train our visual models of what tuples looks like. The VAL tuples will be used to learn how much visual alignment is weighted relative to the textual alignment. The TEST tuples will be used to evaluate the performance of our approach.

Note that we do not collect illustrations for each VAL *tuple* because tuples may contain nouns that our clipart library does not have. Instead, we collect illustrations for each of the VAL *relations*. Workers choose to depict these relations with plausible primary and second objects of their choice, providing an additional source of commonsense knowledge. Regardless, as will be evident in the next section, our model is capable of dealing with nouns and relations at test time that were not present during training.

4. Approach

We first describe our joint text and vision model, followed by a description of the training procedure.

4.1. Model

Let us start by laying out some notation. We are given a commonsense assertion $t' = (t'_P, t'_R, t'_S)$ at test time, whose plausibility is to be evaluated. t'_P is the primary noun, t'_R is the relation, and t'_S is the secondary noun. For each abstract training scene created by AMT workers $i \in I$ we are given the primary and secondary clipart objects c^i_P and c^i_S, as well as a tuple $t^i = (t^i_P, t^i_R, t^i_S)$ containing the names of the primary and secondary objects (nouns), and the relation they participate in. Thus, a training instance i is represented by $\Omega^i = \{c^i_P, c^i_S, t^i\}$.

We score the plausibility of test tuple t' using the following linear scoring function:

$$score(t') = \alpha \cdot f_{text}(t') + \beta \cdot f_{visual}(t') \quad (1)$$

Where α and β tradeoff the weights given to the text alignment score f_{text} and the vision alignment score f_{vision} respectively. The text and vision alignment scores estimate how well the test tuple t' aligns to all training instances – both textual (TRAIN tuples provided by AMT workers) and visual (training abstract scenes provided by AMT workers). Tuples which align well with known (previously seen and/or read) concepts are considered to be more plausible.

Vision and text alignment functions: Both our vision and text alignment functions have the following form:

$$f(t') = \frac{1}{|I|} \sum_{i \in I} \max(h(t', \Omega^i) - \delta, 0) \quad (2)$$

Where f can be either f_{text} or f_{vision}. The average goes over all training instances (i.e., abstract scenes with associated annotated tuples) in our training set. The activation of a training instance with respect to a test tuple is determined by h, which has different forms for vision and text. A ReLU (Rectified Linear Unit) function is applied to the activation score offset by δ. We use a threshold of zero for the ReLU because the notion of negative plausibility evidence for a tuple is not intuitive. One can view Equation 2 as counting how many times a tuple was observed during training. The parameter δ is used to threshold the activation h to estimate counts. From here on we refer to h as the alignment score (overloaded with f).

Text alignment score: The textual alignment score h_{text} between two tuples is a linear combination of similarities between the corresponding pairs of primary nouns, relations, and secondary nouns. These similarities are computed using dot products in the word2vec embedding space [27]. For nouns or relations containing more than one word (e.g., "gather around" or "chair legs"), we average the word2vec vectors of each word to obtain a single vector.

Let $W(x)$ be the vector space embedding of a noun or relation x. The text alignment score is given as follows:

$$h_{text}(t', \Omega^i) = W(t'_P)^T \cdot W(t^i_P) + W(t'_R)^T \cdot W(t^i_R) + W(t'_S)^T \cdot W(t^i_S) \quad (3)$$

Where $\cdot$ denotes the cosine similarity between vectors.

Vision alignment score: The visual alignment score computes the alignment between (i) a given test tuple and (ii) the pair of clipart pieces selected by AMT workers as being the primary and secondary objects in a training instance i. It measures how well the pair of clipart pieces (c^i_P, c^i_S) depict the test tuple t'. If a test tuple finds support from a large number of visual instances, it is likely to be plausible. Note that we are measuring similarity between words and arrangements of clipart pieces. Consequently, this is a multimodal similarity function.

Given the pair of primary and secondary clipart pieces annotated in training instance Ω^i, we extract features as described in Section 5. We denote these extracted features as $u(c^i_P, c^i_S)$. Using these visual features from the training instance Ω^i and text embeddings from test tuple t', we compute the following vision alignment score:

$$h_{vision}(t', \Omega^i) = u(c^i_P, c^i_S)^T A_P W(t'_P) + u(c^i_P, c^i_S)^T A_R W(t'_R) + u(c^i_P, c^i_S)^T A_S W(t'_S) \quad (4)$$

Where A_P, A_R, and A_S are alignment parameters to be learnt. Our vision alignment score measures how well the t'_P, t'_R, and t'_S individually match the visual features $u(c^i_P, c^i_S)$ that describe a pair of clipart objects in training instance Ω_i. One can think of $u(c^i_P, c^i_S)A_P$, $u(c^i_P, c^i_S)A_R$, and $u(c^i_P, c^i_S)A_S$ as embeddings or projections from the vision space to the word2vec text space, such that a high dot product in word2vec space leads to high alignment, and subsequently a high plausibility score for plausible tuples. The embeddings are learnt separately for t'_P, t'_R and t'_S (as parameterized by A_P, A_R and A_S) because different visual features might be useful for aligning to the primary noun, relation, and secondary noun.

The parameters A_P, A_R, and A_S can also be thought of as grounding parameters. That is, given a word2vec vector W, we learn parameters to find the visual instantiation of W. $A_R W(t'_R)$ can be thought of as the visual instantiation of t'_R which captures what the interaction between two objects related by relation t'_R looks like. $A_P W(t'_P)$ and $A_S W(t'_S)$ can be thought of as identifying which clipart pieces and with what attributes correspond to nouns t'_P and t'_S. Our model finds the visual grounding of t'_P, t'_R, and t'_S separately, and then measures similarity of the inferred grounding to the actual visual features observed in training instances. Thus, given a test tuple, we *hallucinate* a grounding for it and measure similarity of the hallucination with the training data. Note that these hallucinations are learnt

discriminatively to help us align concepts in vision and text such that plausible tuples are scored highly.

4.2. Training

To learn the parameters A_P, A_R, A_S in our vision alignment scoring function (Equation 4), we consider the outer product space of the vectors u and W. We learn a linear SVM in this space to separate the training instances (tuples + corresponding abstract scenes, Section 3.3), from a set of negatives. Each negative instance is a tuple from our TRAIN set, paired with a random abstract scene from our training data. We sample three times as many negatives as positives. Overall we have 4260 positives and 12780 negatives. Finally, the learnt vectors are reshaped to get A_P, A_R and A_S respectively. We learn the vision vs. text tradeoff parameters α and β (Equation 1) on the VAL set of tuples (Section 3.2). Recall that these include seed and expanded tuples, along with annotations indicating which tuples are plausible and which are not. We use the vision and text alignment scores as features and train a binary SVM to separate plausible tuples from implausible ones. The weights learnt by the SVM correspond to α and β. Finally, the parameter δ in Equation 2 is set using grid search on the VAL set to maximize the average precision (AP) of predicting a tuple as being plausible (positive) or not.

5. Experimental Setup

We first describe the features we extract from the abstract scenes. We then list the baselines we compare to.

5.1. Visual Features

As explained in Section 3.1, we have annotations indicating which pairs of objects (c_P, c_S) in an abstract scene participated in the corresponding annotated tuple. Using these objects and the remaining scene, we extract three kinds of features to describe the pair of objects (c_P, c_S): 1) Object Features 2) Interaction Features 3) Scene Features. These three together form our visual feature set. **Object Features** consist of the type (category, instance) of the object (Section 3.1), flip (left facing or right) of the object, absolute location, attributes (for humans), and poses (for humans and animals). The absolute location feature is modeled using a Gaussian Mixture Model (GMM) with 9 components, learnt separately across five discrete depth levels, similar to [38]. The GMM components are common across all objects, and are learnt using all objects present in all abstract scenes. Human attributes are age (5 discrete values), skin color (3 discrete values) and gender (2 discrete values). Animals have 5 discrete poses. Human pose features are constructed using keypoint locations. These include global, contact, and orientation features [2]. Global features measure the position of joints with respect to three gaussians placed on the head, torso, and feet respectively. Contact features place smaller gaussians at each joint and measure the positions of other joints with respect to each joint. Orientation features measure the joint angles between connected keypoints. **Interaction Features** encode the relative locations of the two objects participating in the relation, normalized for the flip and depth of the first object. This results in the relative location features being asymmetric. We compute the relative location of the primary object relative to the secondary object and vice versa. Relative locations are encoded using a 24 component GMM (similar to [38]). **Scene Features** indicate which types (category, instance) of objects (other than c_P and c_S) are present in the scene. Overall, there are 493 object features each for the primary and secondary objects, 48 interaction features, and 188 global features, resulting in a visual feature vector of dimension 1222.

5.2. Baselines

We experiment with a variety of strong baselines that use text information alone. They help evaluate how much complementary information vision adds, and if this additional information can be obtained simply from additional or different kinds of text (e.g., generic vs. visual text).

- **WikiEmbedding**: Our first baseline uses the f_{text} part of our model (Equation 1) alone. It uses word2vec trained on generic Wikipedia text.
- **COCOEmbedding**: Our next baseline also uses the f_{text} part of our model (Equation 1) alone, but uses word2vec trained on visual text (>400k captions in the MS COCO training dataset).
- **ValText**: Recall that both our TEST and VAL tuples were extracted from captions describing COCO images. Our next baseline computes the plausibility of a test tuple by counting how often that tuple occurred in VAL. This helps assess the overlap between our TEST and VAL tuples (recall: no images are shared between TEST and VAL). Note that the above two baselines, WikiEmbedding and COCOEmbedding, can be thought of as ValText but by using soft similarities (in word2vec space) rather than using counts based on exact matches.
- **LargeVisualText**: Our next baseline is a stronger version of ValText. Instead of using just our VAL tuples to evaluate the plausibility of a test tuple, it extracts tuples from a large corpus of text describing images (>400k captions in the MS COCO training dataset which are not in our test set (Section 3.2)). This gives us a set of 91K assertions. At test time, we check how many times the test assertion occurred in this set, and use that count as the plausibility score of the test tuple.
- **BigGenericText (Bing)**: In this baseline, we evaluate the performance of assessing the plausibility of tuple $t' = (t'_P, t'_R, t'_S)$ in the test set using all the text on the web. We query the Bing[4] search API and compute the

[4] http://www.bing.com/

Approach	Test Performance	
	AP	Rank Correlation × 100
WikiEmbedding	68.4	41.7
COCOEmbedding	72.2	49.0
ValText	53.0	31.0
LargeVisualText	58.0	37.6
BigGenericText (Bing)	44.6	20.3

Table 1: Performance of different text based methods

Approach	Test Performance	
	AP	Rank Correlation × 100
Text (COCOEmbedding) + Vision	73.6	50.0
Vision Only	68.7	45.3
Text (COCOEmbedding) Only	72.2	49.0

Table 2: Text+ vision outperforms text alone.

log-frequencies of t'_P, t'_R, t'_S as well as t'. We train an SVM on these four features to separate plausible tuples in our VAL set from implausible tuples, and use this SVM at test time to compute the plausibility score of a test tuple.

5.3. Evaluation

Recall that we collected 10 human judgements for the plausibility of each test tuple (Section 3.2). We count the number of subjects who thought the tuple was plausible ($count_+$). We also count the number of subjects who thought the tuple was not plausible ($count_-$). $count_+$ + $count_-$ need not be 10 because subjects were allowed to marked tuples as "does not make sense". These scores are then combined into a single $score = count_+ - count_-$. We threshold these scores at 0 to get our set of positive and negative human (ground truth) labels. That is, a tuple is considered to be plausible if more people thought it is plausible than not. Our method as well as the baselines produce a score for the plausibility of each tuple in the TEST set. These scores are thresholded and compared to the human labels to compute average precision (AP). We also rank tuples based on their predicted plausibility scores and human plausibility scores ($score = count_+ - count_-$). These rankings are compared using a rank correlation, which forms our second evaluation metric.

6. Results

We begin by comparing our text-based baseline models. We then demonstrate the advantage of using vision and text jointly, over using text alone or vision alone. We then show qualitative results. We finally comment on the potential our approach has to enrich existing knowledge bases.

6.1. Different Text Models

Of all the text-alone baselines (Table. 1), we find that BigGenericText (Bing) does the worst, likely because it suffers heavily from the reporting bias on the web. The LargeVisualText baseline does better than Bing, presumably because the captions in MS COCO describe what is seen in the images which may often be mundane details depicted in the image, and aligns well with the source of our tuples (visual text). ValText performs worse than LargeVisualText because ValText uses less data. But adding soft similarities using word2vec embeddings (WikiEmbedding and COCOEmbedding) significantly improves performance (15.4 and 19.2 in absolute AP). COCOEmbedding performs the best among all text-alone baselines, and is what we will use as our "text only" model moving forward.

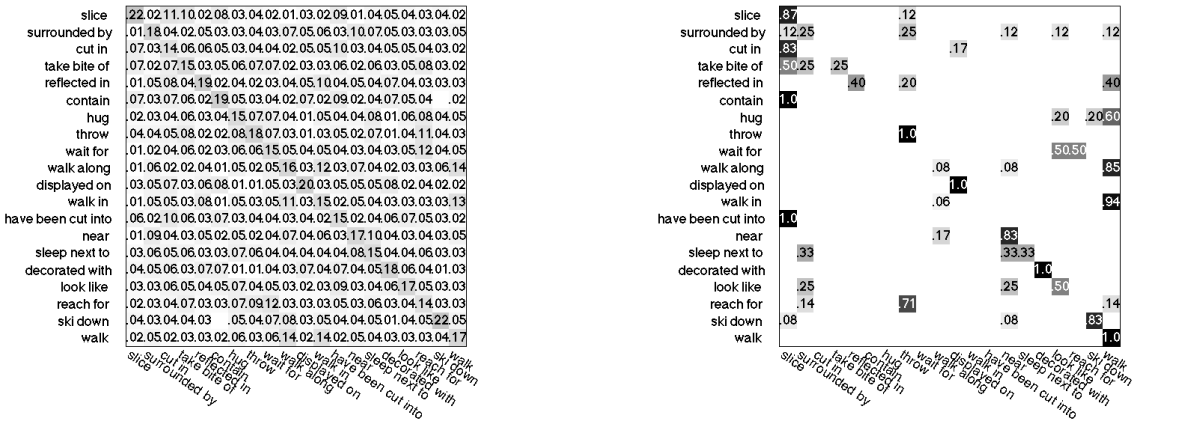
6.2. Joint Text + Vision Model

We compare the performance of text + vision, vision alone, and text alone in Table. 2. We observe that text + vision performs better than text alone and vision alone by 1.4% and 4.9% AP respectively. In terms of rank correlation, text + vision provides an improvement of 1.0 over text alone. Overall, vision and text provide complementary sources of common sense.

6.3. Qualitative Results

We first visualize relation similarity matrices for text and vision alone (Figure 4). Each entry in the text matrix is the word2vec similarity between the relations specified in the corresponding row and columns. Each row is normalized to sum to 1. For vision, each entry in the matrix in the proportion of images depicting a relation (row) whose embeddings – after being transformed by A_R – are most similar to the word2vec representation of another relation (column). This illustrates what our visual alignment function has learnt. We randomly sample a subset of 20 relations for visualization purposes. We can clearly see that the two matrices are qualitatively different and complementary. For instance, visual cues tell us that the relations like "sleep next to" and "surrounded by" are similar.

In Figure 5 we show you several scenes created by AMT workers. Note that for clarity we only show the primary and secondary objects as identified by workers, but our approach uses all objects present in the scene. For each scene, we show the "GT" tuple provided by workers, as well as the "Vision only" tuple. This is computed by embedding the scene using our learnt A_P, A_R, and A_S into the word2vec space and identifying the nouns and relations that are most similar. The left most column shows scenes where the visual prediction matches the GT. The next column shows scenes where the visual prediction is incorrect, but reasonable (even desirable) and would not be captured by text. Consider (boy, hold onto, pizza) and (boy, take, pizza) whose similarity would be difficult to capture via text. The next column shows examples where the tuples are visually as well as textually similar. The last column shows



(a) Textual similarity between relations

(b) Visual similarity between relations

Figure 4: Visual and textual similarities are qualitatively different, and capture complimentary signals.

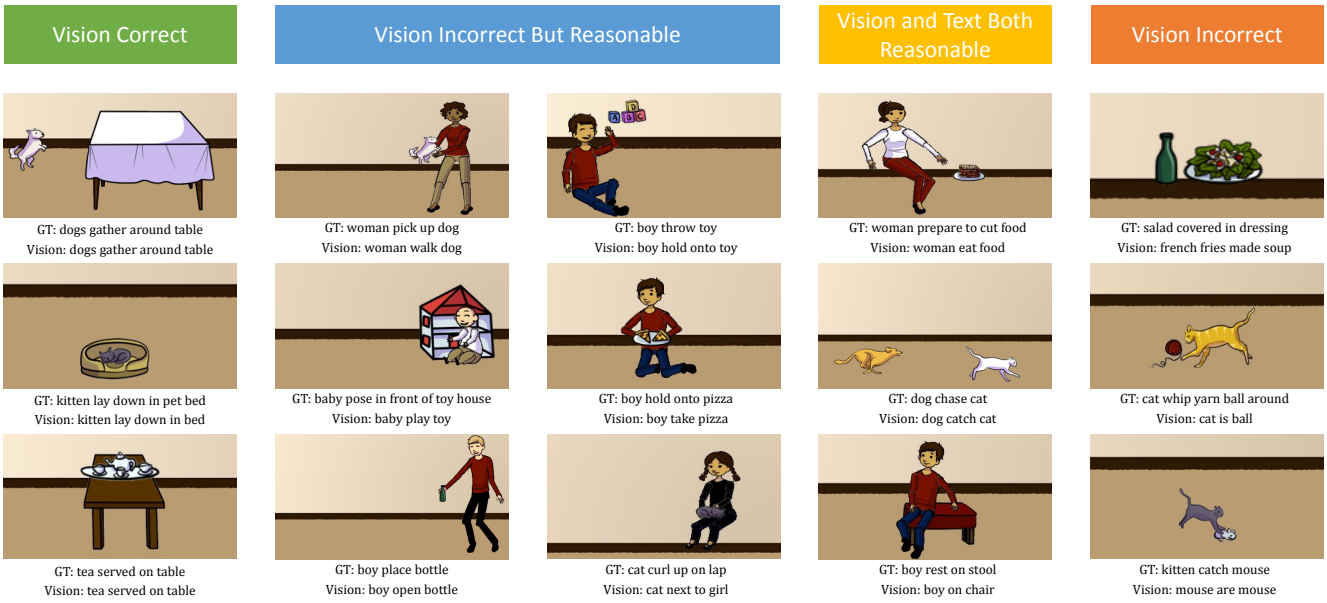


Figure 5: Qualitative examples demonstrating visual similarity between tuples.

failure cases where the visual prediction is unreasonable.

6.4. Enriching Knowledge Bases

ConceptNet [34] contains commonsense knowledge contributed by volunteers. It represents concepts with nodes and relations as edges between them. Out of our 213 VAL relations, only one relation ("made of") currently exists in ConceptNet. Thus, our approach can add many visual commonsense relations to ConceptNet, and boost its recall.

7. Conclusion

In this paper we considered the task of classifying commonsense assertions as being plausible or not based on how similar they are to assertions that are known to be plausible. We argued that vision provides a complementary source of commonsense knowledge to text. Hence, in addition to reasoning about the similarity between tuples based on text, we propose to ground commonsense assertions in the visual world and evaluate similarity between assertions using visual features. We demonstrate the effectiveness of abstract scenes in providing this grounding. We show that assertions can be classified as being plausible or not more accurately using vision + text, than by using text alone. All our datasets and code are publicly available.

Acknowledgements: We thank Stanislaw Antol for his help with the tuple illustration interface. This work is supported in part by an Allen Distinguished Investigator Award from the Paul G. Allen Family Foundation and by a Google Faculty Research Award to D. P.

References

[1] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. VQA: visual question answering. *CoRR*, abs/1505.00468, 2015. 4

[2] S. Antol, C. L. Zitnick, and D. Parikh. Zero-shot learning via visual abstraction. In *ECCV*, 2014. 3, 6

[3] A. C. Berg, T. L. Berg, H. Daume, J. Dodge, A. Goyal, X. Han, A. Mensch, M. Mitchell, A. Sood, K. Stratos, and K. Yamaguchi. Understanding and predicting importance in images. In *CVPR*, 2012. 2

[4] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *ACM SIGMOD*, pages 1247–1250. ACM, 2008. 1, 2

[5] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka Jr, and T. M. Mitchell. Toward an architecture for never-ending language learning. In *AAAI*, 2010. 1, 2

[6] Y.-W. Chao, Z. Wang, R. Mihalcea, and J. Deng. Mining semantic affordances of visual object categories. In *CVPR*, 2015. 2

[7] X. Chen, A. Shrivastava, and A. Gupta. Neil: Extracting visual knowledge from web data. In *ICCV*, 2013. 2

[8] R. Davis, H. Shrobe, and P. Szolovits. What is a knowledge representation? *AI magazine*, 14(1):17, 1993. 1

[9] S. Divvala, A. Farhadi, and C. Guestrin. Learning everything about anything: Webly-supervised visual concept learning. In *CVPR*, 2014. 2

[10] S. K. Divvala, D. Hoiem, J. H. Hays, A. A. Efros, and M. Hebert. An empirical study of context in object detection. In *CVPR*, 2009. 2

[11] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *ACM SIGKDD*, pages 601–610. ACM, 2014. 2

[12] O. Etzioni, A. Fader, J. Christensen, S. Soderland, and M. Mausam. Open information extraction: The second generation. In *IJCAI*, volume 11, pages 3–10, 2011. 2, 3

[13] D. Ferrucci, E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A. A. Kalyanpur, A. Lally, J. W. Murdock, E. Nyberg, J. Prager, N. Schlaefer, and C. Welty. Building watson: An overview of the deepqa project. *AI magazine*, 31(3):59–79, 2010. 2

[14] D. F. Fouhey, V. Delaitre, A. Gupta, A. A. Efros, I. Laptev, and J. Sivic. People watching: Human actions as a cue for single view geometry. In *ECCV*, 2012. 2

[15] D. F. Fouhey and C. L. Zitnick. Predicting object dynamics in scenes. In *CVPR*, 2014. 3

[16] J. Gordon and B. Van Durme. Reporting bias and knowledge acquisition. In *Proceedings of the 2013 Workshop on Automated Knowledge Base Construction*, AKBC '13, pages 25–30, New York, NY, USA, 2013. ACM. 1

[17] H. Grabner, J. Gall, and L. Van Gool. What makes a chair a chair? In *CVPR*, 2011. 2

[18] A. Gupta and L. S. Davis. Beyond nouns: Exploiting prepositions and comparative adjectives for learning visual classifiers. In *ECCV*, 2008. 2

[19] J. Hays and A. A. Efros. Im2gps: estimating geographic information from a single image. In *CVPR*, 2008. 2

[20] V. Hedau, D. Hoiem, and D. Forsyth. Recovering free space of indoor scenes from a single image. In *CVPR*, 2012. 2

[21] J. Hoffart, F. M. Suchanek, K. Berberich, and G. Weikum. Yago2: a spatially and temporally enhanced knowledge base from wikipedia. *Artificial Intelligence*, 194:28–61, 2013. 1

[22] J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. A. Shamma, M. Bernstein, and L. Fei-Fei. Image retrieval using scene graphs. In *CVPR*, 2015. 2

[23] A. Khosla, B. An, J. J. Lim, and A. Torralba. Looking beyond the visible scene. In *CVPR*, 2014. 2

[24] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, and C. Bizer. DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal*, 2014. 1

[25] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*. 2014. 3

[26] X. Lin and D. Parikh. Don't just listen, use your imagination: Leveraging visual common sense for non-visual tasks. In *CVPR*, 2015. 2, 3

[27] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013. 2, 5

[28] G. A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995. 1

[29] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. *EMNLP*, 12, 2014. 2

[30] L. Pickup, Z. Pan, D. Wei, Y. Shih, C. Zhang, A. Zisserman, B. Scholkopf, and W. Freeman. Seeing the arrow of time. In *CVPR*, 2014. 2

[31] H. Pirsiavash, C. Vondrick, and A. Torralba. Inferring the why in images. *CoRR*, abs/1406.5472, 2014. 2

[32] F. Sadeghi, S. K. Divvala, and A. Farhadi. Viske: Visual knowledge extraction and question answering by visual verification of relation phrases. In *CVPR*, pages 1456–1464, 2015. 2

[33] P. Singh, T. Lin, E. T. Mueller, G. Lim, T. Perkins, and W. L. Zhu. Open mind common sense: Knowledge acquisition from the general public. In *On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE*, pages 1223–1237. Springer, 2002. 1

[34] R. Speer and C. Havasi. Conceptnet 5: A large semantic network for relational knowledge. In *The Peoples Web Meets NLP*, pages 161–176. Springer, 2013. 1, 8

[35] Y. Zhu, A. Fathi, and L. Fei-Fei. Reasoning about object affordances in a knowledge base representation. In *ECCV*, 2014. 2

[36] C. Zitnick, R. Vedantam, and D. Parikh. Adopting abstract images for semantic scene understanding. *PAMI*, 2014. 3

[37] C. L. Zitnick and D. Parikh. Bringing semantics into focus using visual abstraction. In *CVPR*, 2013. 3

[38] C. L. Zitnick, D. Parikh, and L. Vanderwende. Learning the visual interpretation of sentences. In *ICCV*, 2013. 3, 6