Learning commonsense knowledge is a problem at the core of Artificial Intelligence. 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. However, while unwritten, this commonsense knowledge is not unseen. Our visual world is replete with structure modeled by commonsense knowledge. In this work we explore how this structure could be utilized to model common sense better.

Extracting commonsense knowledge from visual content requires automatic and accurate detection of objects, their attributes, poses and interactions. These remain key challenges in computer vision. Our 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. Thus, we make use of human generated abstract scenes made from clipart for learning common sense. More specifically, we consider the task of assessing the plausibility of an interaction or relation between a pair of objects. We consider common sense tuples or assertions \((t)\) of the kind (primary object – \(t_p\), relation – \(t_R\), secondary object – \(t_S\)) e.g., (boy, kicks, ball).

Given a tuple to process, our model measures its alignment to a set of concepts known to be plausible. A baseline approach would be to use text-based similarity to compute this alignment. In addition to text, we propose to ground common sense assertions into the visual world and evaluate similarity between assertions using visual features from abstract scenes (Figure 1).

**Data** We extract our commonsense tuples using the ReVerb [1] information extraction tool on sentences from the MSCOCO [2] dataset. We do some post processing on the tuples to “clean” them. We also sample random objects \((t_p, t_S)\) for each relation \(t_R\) to generate extra commonsense tuples. Supervision on the set of extracted and sampled assertions tells us which are plausible and which are implausible. We model a large number of free form relations (213) and nouns (2466), which may form over \(\approx 1\) billion tuples. Overall, our TEST set contains 204 relations and 14,332 assertions while our VAL (validation) set contains 213 relations and 14,548 assertions. The VAL set is used to pick hyperparameters for combining vision and text in our model.

It is important for the library of clipart pieces to be expressive enough to model such a large number of relations. While previous works using visual abstractions [5] depicted a boy and a girl playing in a park, our clipart library contains 20 “paper-doll” human models spanning genders, races and ages with 8 different expressions. Overall, it contains over 100 small and large objects and 31 animals in various poses. Our clipart is also more realistic looking than previous work. We use this vocabulary to collect training scenes to learn our visual alignment model.

Our TRAIN (training) set is made of abstract scenes depicting the 213 VAL relations on Amazon Mechanical Turk (AMT). Workers are shown an existing background scene and asked to modify it to contain the relation of interest. 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.

**Model** We are given a commonsense assertion \(t' = (t_p, t_R, t_S)\) at test time, whose plausibility is to be evaluated. We score the plausibility of using a linear combination of text and visual alignment functions. We compute alignment with respect to tuples \((t')\) and cliparts \((c_p, c_R, c_S)\) belonging to the training illustration \(\Omega\). We then sum the alignments across all training instances to get the alignment score. The alignment functions can be interpreted as counting soft-occurrences of the test tuple in the training set. We use the popular word2vec [3] vector space embeddings to compute the text alignment. We then compute the vision alignment score as follows:

\[
h_{vision}(t', \Omega) = u(c_p, c_R)^T A_p W(t_p) + u(c_p, c_S)^T A_R W(t_R') + u(c_p, c_S)^T A_S W(t_S')
\]

Where \(A_p, A_R, \text{and } A_S\) are alignment parameters to be learnt. Our vision alignment score measures how well the \(t_p, t_R', \text{and } t_S'\) individually match the visual features \(u(c_p, c_R')\) that describe a pair of clipart objects in training instance \(\Omega\). One can think of \(u(c_p, c_R')\) 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 \(t_p, t_R', \text{and } t_S'\) (as parameterized by \(A_p, A_R, \text{and } A_S\) because different visual features might be useful for aligning to the primary noun, relation, and secondary noun. The parameters \(A_p, A_R, \text{and } A_S\) can also be thought of as grounding parameters. That is, given a word2vec vector \(W\), we learn parameters to find the visual instantiation of \(W\). \(A_p W(t_p')\) can be thought of as the visual instantiation of \(t_p'\) 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 \(t_p, t_R', \text{and } t_S'\) 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.

---

**Learning Commonsense Knowledge Through Visual Abstraction**

Ramakrishna Vedantam1, Xiao Lin1, Tanmay Batra2, C. Lawrence Zitnick3, Devi Parikh1

1Virginia Tech 2Carnegie Mellon University 3Microsoft Research

---

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.
Training   To learn the parameters $A_P$, $A_R$, $A_S$ in our vision alignment scoring function (Equation 1), 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), from a set of negatives (randomly sampled). Finally, the learnt vectors are reshaped to get $A_P$, $A_R$ and $A_S$ respectively. The parameters to combine the text and vision alignment scores are learnt discriminatively on the VAL set.

Results   The output of our models is a list of tuples in the TEST set with scores indicating their plausibility. Recall that we collect the plausibility ground truth for TEST tuples from AMT. Using them we evaluate the results using Average Precision (AP) and rank correlation as metrics. We try different choices of embeddings for text alignment (shown in Table 1). We experiment with Wikipedia and MSCOCO sentences to learn embeddings. Since our tuples are derived from visual text, using embeddings derived from sentences describing images (COCOEmbedding) helps the task a lot (Table 1). We also evaluate the performance by querying the Bing search API and compute plausibility using tuple counts, which does much worse, suggesting that word frequencies on the web might be unsuitable to assess plausibility. The combined vision + text model described above obtains a boost of 1.4% over purely using text (Table 2).

Qualitative Results   In Figure 2 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 failure cases where the visual prediction is unreasonable.

Enriching Knowledge Bases   ConceptNet [4] 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.

Conclusion   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.

Reference