Combining Knowledge and Reasoning through Probabilistic Soft Logic for Image Puzzle Solving

Somak Aditya, Yezhou Yang, Chitta Baral
School of Computing, Informatics and Decision Systems Engineering
Arizona State University
{saditya1,yz.yang,chitta}@asu.edu

Yiannis Aloimonos
UMIACS, Computer Science
University of Maryland, College Park
yiannis@cs.umd.edu

Abstract

The uncertainty associated with human perception is often reduced by one's extensive prior experience and knowledge. Current datasets and systems do not emphasize the necessity and benefit of using such knowledge. In this work, we propose the task of solving a genre of image-puzzles ("image riddles") that require both capabilities involving visual detection (including object, activity recognition) and, knowledge-based or commonsense reasoning. Each puzzle involves a set of images and the question "what word connects these images?". We compile a dataset of over 3k riddles where each riddle consists of 4 images and a groundtruth answer. The annotations are validated using crowd-sourced evaluation. We also define an automatic evaluation metric to track future progress. Our task bears similarity with the commonly known IQ tasks such as analogy solving, sequence filling that are often used to test intelligence. We develop a Probabilistic Reasoning-based approach that utilizes commonsense knowledge about words and phrases to answer these riddles with a reasonable accuracy. Our approach achieves some promising results for these riddles and provides a strong baseline for future attempts. We make the entire dataset and related materials publicly available to the community (bit.ly/22f9Ala).

1 INTRODUCTION

Human visual perception is greatly aided by the human's knowledge and reasoning (with that knowledge) about the domain (what it is looking at) and purpose (what it is looking for and why) (Lake et al., 2016). This knowledge greatly helps in overcoming the uncertainty often associated with perception. Most work in computer vision do not take into account the vast body of knowledge that humans use in their visual perception. Several researchers[1] have recently pointed out the necessity and the potential benefit of using such knowledge, as well as the lack of it in current systems. This absence is also reflected in the various popular data sets and benchmarks. Our goal in this paper is to present a new task, a corresponding new data set, and our approach to them that highlights the importance of using knowledge and reasoning in visual perception. This necessitates considering issues such as what kind of knowledge is needed, where and how to get them, and what kind of reasoning mechanism to adopt for such knowledge.

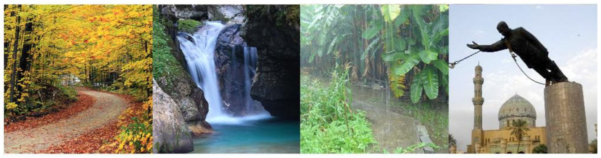

Figure 1: An Image Riddle Example. Question: "What word connects these images?" .

The new task we propose in this paper is referred to as Image Riddles which requires deep conceptual understanding of images. In this task a set of images are provided and one needs to find a common concept that is invoked by all the images. Often the common concept is not something that even a human can observe in the first glance; but after some thought about the images, he/she can come up with it. Hence the word "riddle" in the phrase "image riddles". Figure 1 shows an example

[1] Lake et al. (2016) quotes a reviewer: "Human learners - unlike DQN and many other deep Learning systems - approach new problems armed with extensive prior experience.". The authors also ask "How do we bring to bear rich prior knowledge to learn new tasks and solve new problems?". In "A Path to AI", Prof. Yann Lecun recognizes the absence of common-sense to be an obstacle to AI.

of an image riddle. The images individually connect to multiple concepts such as: *outdoors, nature, trees, road, forest, rainfall, waterfall, statue, rope, mosque* etc. On further thought, the common concept that emerges for this example is “fall”. Here, the first image represents the fall season (*concept*). There is a “waterfall” (*region*) in the second image. In the third image, it shows “rainfall” (*concept*) and the fourth image depicts that a statue is “fall”ing (*action/event*). The word “fall” is invoked by all the images as it shows logical connections to objects, regions, actions or concepts specific to each image. Additionally, the answer also connects the most salient aspects of the images. Other possible answers like “nature” or “outdoors” do not demonstrate such properties. They are too abstract. In essence, answering Image Riddles is a challenging task that not only tests an intelligent agent’s ability to detect visual concepts, but also tests its (ontological) knowledge and its ability to think and reason.

Image Riddles can also be thought of as a visual counterpart to IQ tests such as sequence filling $(x_1, x_2, x_3, ?)$ and analogy solving $(x_1 : y_1 :: x_2 : ?)$[2], where one needs to find commonalities between items. It is worth to note that this task is different from traditional Visual Question-Answering (VQA), as in VQA the queries provide some clues regarding what to look for in the image. Most Image Riddles require both superior detection and reasoning capabilities, whereas a large percentage of questions from the VQA dataset tests mainly the system’s detection capabilities. Moreover, answering Image Riddles differs from both VQA and Captioning in that it requires analysis of multiple seemingly different images.

Hence, this task of answering Image Riddles is simple to explain; shares similarities with well-known and predefined types of IQ questions and it requires a combination of vision and reasoning capabilities. In this paper, we introduce a novel benchmark for Image Riddles and put forward a promising approach to tackle it. In our approach, we first use the state-of-the-art Image Classification techniques (Sood (2015) and He et al. (2016)) to get the top identified class-labels from each image. Given these detections, we use ontological and commonsense relations of these words to infer a set of most probable concepts. We adopt ConceptNet 5 (Liu and Singh, 2004) as the source of commonsense and background knowledge that encodes the relations between words and short phrases through a structured graph. Note, the possible range of candidates are **the entire vocabulary** of ConceptNet 5 (roughly 0.2 million), which is fundamentally different from supervised end-to-end models. For representation and reasoning with this huge probabilistic knowledge one needs a powerful reasoning engine. Here, we adopt the Probabilistic Soft Logic (PSL) (Kimmig et al., 2012; Bach et al., 2013) framework. Given the inferred concepts of each image, we adopt a second stage inference to output the final answer.

Our **contributions** are threefold: i) we introduce the 3K Image Riddles Dataset; ii) we present a probabilistic reasoning approach to solve the riddles with reasonable accuracy; iii) our reasoning module inputs detected words (a closed set of class-labels) and *logically* infers all relevant concepts (belonging to a much larger vocabulary), using background knowledge about words.

2 RELATED WORK

The problem of Image Riddles has some similarities to the genre of topic modeling (Blei, 2012) and Zero-shot Learning (Larochelle et al., 2008). However, this dataset imposes a few unique challenges: i) the possible set of target labels is the entire natural language vocabulary; ii) each image, when grouped with different sets of images can map to a different label; iii) almost all the target labels in the dataset are unique (3k examples with 3k class-labels). These challenges make it hard to simply adopt topic model-based or Zero-shot learning-based approaches.

Our work is also related to the field of **Visual Question Answering** (VQA). Very recently, researchers spent a significant amount of efforts on both creating datasets and proposing new models (Antol et al., 2015; Malinowski et al., 2015; Gao et al., 2015; Ma et al., 2016) for VQA. Interestingly both (Antol et al., 2015; Goyal et al., 2017) and Gao et al. (2015) adapted MS-COCO (Lin et al., 2014) images and created an open domain dataset with human generated questions and answers. Both Malinowski et al. (2015) and Gao et al. (2015) use recurrent networks to encode the sentence and output the answer.

Even though some questions from Antol et al. (2015) and Gao et al. (2015) are very challenging, and actually require logical reasoning in order to answer correctly, popular approaches still aim to learn the direct signal-to-signal mapping from image and question to its answer, given a large enough annotated data. The necessity of common-sense reasoning is often neglected. Here we introduce the new Image Riddle problem to serve as the testbed for vision and reasoning research.

3 KNOWLEDGE AND REASONING MECHANISM

In this Section, we briefly introduce the kind of knowledge that is useful for solving Image Riddles and the

[2] Examples are: word analogy tasks (male : female :: king : ?); numeric sequence filling tasks: $(1, 2, 3, 5, ?)$.

kind of reasoning needed. The primary types of knowledge needed are the distributional and relational similarities between words and concepts. We obtain them from analyzing the ConceptNet knowledge base and using Word2Vec. Both the knowledge sources are considered because ConceptNet embodies commonsense knowledge and Word2vec encodes word-meanings.

ConceptNet (Speer and Havasi, 2012), is a multilingual Knowledge Graph, that encodes commonsense knowledge about the world and is built primarily to assist systems that attempts to understand natural language text. The knowledge in ConceptNet is semi-curated. The nodes (called concepts) in the graph are words or short phrases written in natural language. The nodes are connected by edges which are labeled with meaningful relations. For example: *(reptile, IsA, animal), (reptile, HasProperty, cold blood)* are some edges. Each edge has an associated confidence score. Compared to other knowledge-bases such as WordNet, YAGO, NELL (Suchanek et al., 2007; Mitchell et al., 2015), ConceptNet has a more extensive coverage of English language words and phrases. These properties make this Knowledge Graph a perfect source for the required probabilistic commonsense knowledge. We use different methods on ConceptNet, elaborated in the next section, to define similarity between different types of words and concepts.

Word2vec uses the theory of distributional semantics to capture word meanings and produce word embeddings (vectors). The pre-trained word-embeddings have been successfully used in numerous Natural Language Processing applications and the induced vector-space is known to capture the graded similarities between words with reasonable accuracy (Mikolov et al., 2013). Throughout the paper, for word2vec-based similarities, we use the 3 Million word-vectors trained on Google-News corpus (Mikolov et al., 2013).

The similarity between words w_i and w_j with a similarity score w_{ij} is expressed as propositional formulas of the form: $w_i \Rightarrow w_j : w_{ij}$. (The exact formulas, and when they are bidirectional and when they are not are elaborated in the next section.) To reason with such knowledge we explored various reasoning formalisms and found Probabilistic Soft Logic (PSL) (Kimmig et al., 2012; Bach et al., 2013) to be the most suitable, as it can not only handle relational structure, inconsistencies and uncertainty, thus allowing one to express rich probabilistic graphical models (such as Hinge-loss Markov random fields), but it also seems to scale up better than its alternatives such as Markov Logic Networks (Richardson and Domingos, 2006). In this work, we also use different weights for different groundings of the same rule. Even though some work has been done along this line for MLNs (Mittal et al., 2015), implementing those ideas in MLNs to define weights using word2vec and ConceptNet is not straightforward. Learning grounding-specific weights is also difficult as that will require augmentation of MLN syntax and learning.

3.1 PROBABILISTIC SOFT LOGIC (PSL)

Probabilistic soft logic (PSL) differs from most other probabilistic formalisms in that its ground atoms have continuous truth values in the interval [0,1], instead of having binary truth values. The syntactic structure of rules and the characterization of the logical operations have been chosen judiciously so that the space of interpretations with nonzero density forms a convex polytope. This makes inference in PSL a convex optimization problem in continuous space, which in turn allows efficient inference. We now give a brief overview of PSL.

A PSL model is defined using a set of weighted if-then rules in first-order logic. Let $\boldsymbol{C} = (C_1, ..., C_m)$ be such a collection where each C_j is a disjunction of literals, where each literal is a variable y_i or its negation $\neg y_i$, where $y_i \in \boldsymbol{y}$. Let I_j^+ (resp. I_j^-) be the set of indices of the variables that are not negated (resp. negated) in C_j. Each C_j is:

$$w_j : \wedge_{i \in I_j^-} y_i \rightarrow \vee_{i \in I_j^+} y_i, \tag{1}$$

or equivalently, $w_j : \vee_{i \in I_j^-} (\neg y_i) \bigvee \vee_{i \in I_j^+} y_i$. Each rule C_j is associated with a non-negative weight w_j. PSL relaxes the boolean truth values of each ground atom a (constant term or predicate with all variables replaced by constants) to the interval [0, 1], denoted as $I(a)$. To compute soft truth values, Lukasiewicz's relaxation (Klir and Yuan, 1995) of conjunctions ($\wedge$), disjunctions ($\vee$) and negations ($\neg$) is used:

$$\begin{aligned} I(l_1 \wedge l_2) &= max\{0, I(l_1) + I(l_2) - 1\} \\ I(l_1 \vee l_2) &= min\{1, I(l_1) + I(l_2)\} \\ I(\neg l_1) &= 1 - I(l_1). \end{aligned} \tag{2}$$

In PSL, the ground atoms are considered as random variables and the distribution is modeled using **Hinge-Loss Markov Random Field**, which is defined as follows: Let $\boldsymbol{y}$ and $\boldsymbol{x}$ be two vectors of n and n' random variables respectively, over the domain $D = [0,1]^{n+n'}$. The feasible set $\tilde{D}$ is a subset of D, which satisfies a set of inequality constraints over the random variables. A *Hinge-Loss Markov Random Field* $\mathbb{P}$ is a probability density, defined as: if $(\boldsymbol{y}, \boldsymbol{x}) \notin \tilde{D}$, then $\mathbb{P}(\boldsymbol{y}|\boldsymbol{x}) = 0$; if $(\boldsymbol{y}, \boldsymbol{x}) \in \tilde{D}$, then:

$$\mathbb{P}(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{w}, \boldsymbol{x})} exp(-f_{\boldsymbol{w}}(\boldsymbol{y}, \boldsymbol{x})), \tag{3}$$

where $Z(\boldsymbol{w}, \boldsymbol{x}) = \int_{\boldsymbol{y}|(\boldsymbol{y},\boldsymbol{x}) \in \tilde{D}} exp(-f_{\boldsymbol{w}}(\boldsymbol{y}, \boldsymbol{x})) d\boldsymbol{y}$.

Here, the hinge-loss energy function $f_{\boldsymbol{w}}$ is defined as: $f_{\boldsymbol{w}}(\boldsymbol{y}, \boldsymbol{x}) = \sum_{j=1}^{m} w_j (max\{\boldsymbol{l_j}(\boldsymbol{y}, \boldsymbol{x}), 0\})^{p_j}$, where w_j's are non-negative free parameters and $\boldsymbol{l_j}(\boldsymbol{y},\boldsymbol{x})$ are linear constraints over $\boldsymbol{y},\boldsymbol{x}$ and $p_j \in \{1,2\}$. As we are interested in finding the maximum probable solution given the evidence, the inference objective of HL-MRF becomes:

$$\mathbb{P}(\boldsymbol{y}|\boldsymbol{x}) \equiv \underset{\boldsymbol{y}\in[0,1]^n}{\arg\min} \sum_{j=1}^{m} w_j (max\{\boldsymbol{l_j}(\boldsymbol{y},\boldsymbol{x}), 0\})^{p_j}. \quad (4)$$

In PSL, each logical rule C_j in the database $\boldsymbol{C}$ is used to define $\boldsymbol{l_j}(\boldsymbol{y},\boldsymbol{x})$, i.e. the linear constraints over $(\boldsymbol{y},\boldsymbol{x})$. Given a set of weighted logical formulas, PSL builds a graphical model defining a probability distribution over the continuous value space of the random variables in the model.

More precisely, $\boldsymbol{l_j}(.)$ is defined in terms of "distance to satisfaction". For each rule $C_j \in \boldsymbol{C}$ this "distance to satisfaction" is measured using the term $w_j \times max\{1 - \sum_{i\in I_j^+} y_i - \sum_{i\in I_j^-}(1-y_i), 0\}$. This encodes the penalty if a rule is not satisfied. Then, the right hand side of the Eq. 4 becomes:

$$\underset{\boldsymbol{y}\in[0,1]^n}{\arg\min} \sum_{C_j\in\boldsymbol{C}} w_j\, max\{1 - \sum_{i\in I_j^+} y_i - \sum_{i\in I_j^-}(1-y_i), 0\}, \quad (5)$$

which is used to estimate $\mathbb{P}(\boldsymbol{y}|\boldsymbol{x})$ efficiently.

4 APPROACH

Given a set of images ($\{\mathcal{I}_1,\mathcal{I}_2,\mathcal{I}_3,\mathcal{I}_4\}$), our objective is to determine a set of ranked words (T) based on how well they semantically connect the images. In this work, we present an approach that uses the previously introduced Probabilistic Reasoning framework on top of a probabilistic Knowledge Base (ConceptNet). It also uses additional semantic knowledge from Word2vec. Using these knowledge sources, we predict the answers to the riddles. Although our approach consists of multiple resources and stages, it can be easily modularized, pipelined and reproduced. It is also worth to mention that the PSL engine is a general tool. It could be used for further research along the conjunction of vision, language and reasoning.

4.1 OUTLINE OF OUR FRAMEWORK

As outlined in algorithm 1, for each image $\mathcal{I}_k$ (here, $k \in \{1,...,4\}$), we follow three steps to infer related words and phrases: i) Image Classification: we get top class labels and the confidence from Image Classifier ($\boldsymbol{S_k}, \tilde{P}(\boldsymbol{S_k}|\mathcal{I}_k)$), ii) Rank and Retrieve: using these labels and confidence scores, we rank and retrieve top related words ($\boldsymbol{T_k}$) from ConceptNet ($\mathcal{K}_{cnet}$), iii) Probabilistic Reasoning and Inference (Stage I): using the labels ($\boldsymbol{S_k}$) and the top related words ($\boldsymbol{T_k}$), we design an inference model to logically infer final set of words ($\boldsymbol{\hat{T}_k}$) for each image. Lastly, we use another probabilistic reasoning model (Stage II) on the combined set of inferred words (*targets*) from all images in a riddle. This model assigns the final confidence scores on the combined set of targets (T). We depict the pipeline with an example in Figure 2.

Algorithm 1: Solving Image Riddles

```
procedure UnRiddler($\mathcal{I} = \{\mathcal{I}_1,\mathcal{I}_2,\mathcal{I}_3,\mathcal{I}_4\}, \mathcal{K}_{cnet}$)
    for $\mathcal{I}_k \in \mathcal{I}$ do
        $\tilde{P}(\boldsymbol{S_k}|\mathcal{I}_k)$ = getClassLabelsNeuralNetwork($\mathcal{I}_k$).
        for $s \in \boldsymbol{S_k}$ do
            $\boldsymbol{T_s}, W_m(s,\boldsymbol{T_s})$ = retrieveTargets($s, \mathcal{K}_{cnet}$);
            $W_m(s,t_j) = sim(s,t_j) \forall t_j \in \boldsymbol{T_s}$.
        end for
        $\boldsymbol{T_k}$ = rankTopTargets($\tilde{P}(\boldsymbol{S_k}|\mathcal{I}_k), \boldsymbol{T_{S_k}}, W_m$);
        $I(\hat{T}_k)$ = inferConfidenceStageI($\boldsymbol{T_k}, \tilde{P}(\boldsymbol{S_k}|\mathcal{I}_k)$).
    end for
    $I(T)$ = inferConfidenceStageII($[\boldsymbol{\hat{T}_k}]_{k=1}^4, [\tilde{P}(\boldsymbol{S_k}|\mathcal{I}_k)]_{k=1}^4$).
end procedure
```

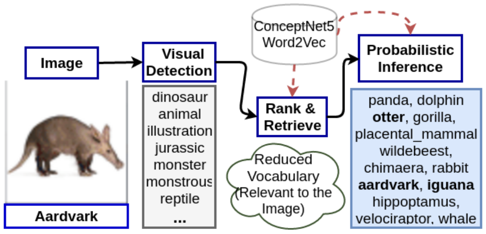

Figure 2: An overview of the framework followed for each Image; demonstrated using an example image of an *aardvark* (resembles animals such as tapir, ant-eater). As shown, the uncertainty in detecting concepts is reduced after considering additional knowledge. We run a similar pipeline for each image and then infer final results using a final Probabilistic Inference Stage (Stage II).

4.2 IMAGE CLASSIFICATION

Neural Networks trained on ample source of images and numerous image classes has been very effective. Studies have found that convolutional neural networks (CNN) can produce near human level image classification accuracy (Krizhevsky et al., 2012), and related work has been used in various visual recognition tasks such as scene labeling (Farabet et al., 2013) and object recognition (Girshick et al., 2014). To exploit these advances, we use the state-of-the-art class detections provided by the Clarifai API (Sood, 2015) and the Deep Residual Network Architecture by (He et al., 2016) (using the trained ResNet-200 model). For each image ($\mathcal{I}_k$) we use top 20 detections ($\boldsymbol{S_k}$) (*seeds*). Figure 2 provides an example. Each detection is accompanied with the classifier's confidence score ($\tilde{P}(\boldsymbol{S_k}|\mathcal{I}_k)$).

4.3 RETRIEVE AND RANK RELATED WORDS

Our goal is to logically infer words or phrases that represent (higher or lower-level) concepts that can best explain the co-existence of the detected *seeds* in a scene. For examples, for "hand" and "care", implied words could be "massage", "ill", "ache" etc. For "transportation" and "sit", implied words/phrases could be "sit in bus" and "sit in plane". The reader might be inclined to infer other concepts. However, to "infer" is to derive "logical" conclusions. Hence, we prefer the concepts which shares strong explainable connections (i.e. relational similarity) with the *seeds*.

A logical choice would be traversing a knowledge-graph like ConceptNet and find the common reachable nodes from these *seeds*. As this is computationally infeasible, we use the association-space matrix representation of ConceptNet, where the words are represented as vectors. The similarity between two words approximately embodies the strength of the connection over all paths connecting the two words in the graph. We get the top similar words for each *seed*, approximating the reachable nodes.

4.3.1 Retrieve Related Words For a Seed

We observe that, for objects, the ConceptNet-similarity gives a poor result (See Table 1). So, we define a metric called **visual similarity**. Let us call the similar words as *targets*. In this metric, we represent the seed and the target as vectors. To define the dimensions, for each *seed*, we use the relations (HasA, HasProperty, PartOf and MemberOf). We query ConceptNet to get the related words (*W1,W2,W3...*) under such relations for the seed-word and its superclasses. Each of these relation-word pairs (i.e. *HasA-W1,HasA-W2,PartOf-W3,...*) becomes a separate dimension. The values for the seed-vector are the weights assigned to the assertions. For each *target*, we query ConceptNet and populate the target-vector using the edge-weights for the dimensions defined by the seed-vector.

To get the top words using visual similarity, we use the cosine similarity of the seed-vector and the target-vector to re-rank the top 10000 retrieved similar target-words. For abstract *seeds*, we do not get any such relations and thus use the ConceptNet similarity directly.

Table 1 shows the top similar words using ConceptNet, word2vec and visual-similarity for the word "men".

Formulation: For each seed (s), we get the top words ($\boldsymbol{T_s}$) from ConceptNet using the visual similarity metric and the similarity vector $W_m(s, \boldsymbol{T_s})$. Together for an image, these constitute $\boldsymbol{T_{S_k}}$ and the matrix W_m, where $W_m(s_i, t_j) = sim_{vis}(s_i, t_j) \forall s_i \in S_k, t_j \in \boldsymbol{T_{S_k}}$.

ConceptNet	Visual Similarity	Word2vec
man, merby, misandrous, philandry, male_human, dirty_pig, mantyhose, date_woman,guyliner,manslut	priest, uncle, guy, geezer, bloke, pope, bouncer, ecologist, cupid, fella	women, men, males, mens, boys, man, female, teenagers,girls,ladies

Table 1: Top 10 similar Words for "Men". The ranked list based on visual-similarity ranks *boy, chap, husband, godfather, male_person, male* in the ranks 16 to 22. See appendix for more.

A large percentage of the error from Image Classifiers are due to visually similar objects or objects from the same category (Hoiem et al., 2012). In such cases, we use this visual similarity metric to predict the possible visually similar objects and then use an inference model to infer the actual object.

4.3.2 Rank Targets

We use the classifier confidence scores $\tilde{P}(\boldsymbol{S_k}|\mathcal{I}_k)$ as an approximate vector representation for an image, in which the *seeds* are the dimensions. The columns of W_m provides vector representations for the target words ($t \in \boldsymbol{T_{S_k}}$) in the space. We calculate cosine similarities for each target with such a image-vector and then re-rank the targets. We denote the top $\boldsymbol{\theta}_{\#t}$ targets as $\boldsymbol{T_k}$ (see Table. 2).

4.4 PROBABILISTIC REASONING AND INFERENCE

4.4.1 PSL Inference Stage I

Given a set of candidate *targets* $\boldsymbol{T_k}$ and a set of weighted *seeds* ($\boldsymbol{S_k}, \tilde{P}(\boldsymbol{S_k}|\mathcal{I}_k)$), we build an inference model to infer a set of most probable *targets* ($\hat{\boldsymbol{T}}_{\boldsymbol{k}}$). We model the joint distribution using PSL as this formalism adopts Markov Random Field which obeys the properties of Gibbs Distribution. In addition, a PSL model is declared using rules. Given the final answer, the set of satisfied rules show the logical connections between the detected words and the final answer. The PSL model can be best explained as an Undirected Graphical Model involving *seeds* (observed) and *targets* (unobserved). We define the seed-target and target-target potentials using PSL rules. We connect each seed to each target and the potential depends on their similarity and the target's popularity bias. We connect each target to $\boldsymbol{\theta}_{t\text{-}t}$ (1 or 2) maximally similar targets. The potential depends on their similarity.

Formulation: Using PSL, we add two sets of rules: i) to define seed-target potentials, we add rules of the form $wt_{ij} : s_{ik} \rightarrow t_{jk}$ for each word $s_{ik} \in \boldsymbol{S_k}$ and target $t_{jk} \in \boldsymbol{T_k}$; ii) to define target-target potentials, for each target t_{jk}, we take the most similar $\boldsymbol{\theta}_{t\text{-}t}$ targets (T_j^{max}).

For each target t_{jk} and each $t_{mk} \in T_j^{max}$, we add two rules $wt_{jm} : t_{jk} \rightarrow t_{mk}$ and $wt_{jm} : t_{mk} \rightarrow t_{jk}$. Next, we describe the choices in detail.

i) From the perspective of optimization, the rule $wt_{ij} : s_{ik} \rightarrow t_{jk}$ adds the term $wt_{ij} * max\{I(s_{ik}) - I(t_{jk}), 0\}$ to the objective. This means that if confidence score of the target t_{jk} is not greater than $I(s_{ik})$ (i.e. $\tilde{P}(\boldsymbol{S_k}|\mathcal{I}_k)$), then the rule is not satisfied and we penalize the model by wt_{ij} times the difference between the confidence scores. We add the above rule for seeds and targets for which the combined similarity (wt_{ij}) exceeds certain threshold $\boldsymbol{\theta}_{sim,psl1}$. We encode the commonsense knowledge of words and phrases obtained from different knowledge sources into the weights of these rules wt_{ij}. It is also important that the inference model is not biased towards more popular targets (i.e. abstract words or words too commonly used/detected in corpus). We compute eigenvector centrality score ($\mathbb{C}(.)$) for each word in the context of ConceptNet. Higher $\mathbb{C}(.)$ indicates higher connectivity of a word in the graph. This yields a higher similarity score to many words and might give an unfair bias to this *target* in the inference model. Hence, the higher the $\mathbb{C}(.)$, the word provides less specific information for an image. Hence, the weight becomes

$$wt_{ij} = \boldsymbol{\theta}_{\alpha_1} * sim_{cn}(s_{ik}, t_{jk}) + \boldsymbol{\theta}_{\alpha_2} * sim_{w2v}(s_{ik}, t_{jk}) + 1/\mathbb{C}(t_{jk}), \qquad (6)$$

where $sim_{cn}(.,.)$ is the normalized ConceptNet-based similarity. $sim_{w2v}(.,.)$ is the normalized word2vec similarity of two words and $\mathbb{C}(.)$ is the eigenvector-centrality score of the argument in the ConceptNet matrix.

ii) To model dependencies among the targets, we observe that if two concepts t_1 and t_2 are very similar in meaning, then a system that infer t_1 should infer t_2 too, given the same set of observed words. Therefore, the two rules $wt_{jm} : t_{jk} \rightarrow t_{mk}$ and $wt_{jm} : t_{mk} \rightarrow t_{jk}$ are designed to force the confidence values of t_{jk} and t_{mk} to be as close to each other as possible. wt_{jm} is the same as Equation 6 without the penalty for popularity.

Using Equation 5, the PSL inference objective becomes:

$$\arg\min_{I(\boldsymbol{T_k}) \in [0,1]^{|T_k|}} \sum_{s_{ik} \in \boldsymbol{S_k}} \sum_{t_{jk} \in \boldsymbol{T_k}} wt_{ij}\ max\{I(s_{ik}) - I(t_{jk}), 0\} + \sum_{t_{jk} \in \boldsymbol{T_k}} \sum_{t_{mk} \in T_j^{max}} wt_{jm} \Big\{ max\{I(t_{mk}) - I(t_{jk}), 0\} + max\{I(t_{jk}) - I(t_{mk}), 0\} \Big\}.$$

To let the targets compete against each other, we add one more constraint on the sum of the confidence scores of the targets i.e. $\sum_{j:t_{jk} \in \boldsymbol{T_k}} I(t_{jk}) \leq \boldsymbol{\theta}_{sum1}$. Here $\boldsymbol{\theta}_{sum1} \in \{1, 2\}$ and $I(t_{jk}) \in [0, 1]$. The above optimizer provides us $\mathbb{P}(\boldsymbol{T_k}|\boldsymbol{S_k})$ and thus the top set of targets $[\hat{\boldsymbol{T}}_{\boldsymbol{k}}]_{k=1}^4$.

4.4.2 PSL Inference Stage II

To learn the most probable common targets jointly, we consider the *targets* and the *seeds* from all images together. Assume that the *seeds* and the *targets* are nodes in a knowledge-graph. Then, the most appropriate target-nodes should observe similar properties as an appropriate answer to the riddle: i) a target-node should be connected to the high-weight seeds in an image i.e. should relate to the important aspects of the image; and ii) a target-node should be connected to seeds from all images.

Formulation: Here, we use the rules $wt_{ij} : s_{ik} \rightarrow t_{jk}$ for each word $s_{ik} \in \boldsymbol{S_k}$ and target $t_{jk} \in \hat{\boldsymbol{T}}_{\boldsymbol{k}}$ for all $k \in \{1, 2.., 4\}$. To let the set of targets compete against each other, we add the constraint $\sum_{k=1}^{4} \sum_{j:t_{jk} \in \hat{\boldsymbol{T}}_{\boldsymbol{k}}} I(t_{jk}) \leq \boldsymbol{\theta}_{s2}$. Here $\boldsymbol{\theta}_{s2} = 1$ and $I(t_{jk}) \in [0, 1]$. The second inference stage provides us $\mathbb{P}([\hat{\boldsymbol{T}}_{\boldsymbol{k}}]_{k=1}^4 | \boldsymbol{S_1}, \boldsymbol{S_2}, \boldsymbol{S_3}, \boldsymbol{S_4})$ and thus the top targets that constitutes the final answers. To minimize the penalty for each rule, the optimal solution maximizes the confidence score of t_{jk}. To minimize the overall penalty, it should maximize the confidence scores of these targets which satisfy most of the rules. As the summation of confidence scores is bounded, only a few top inferred targets should have non-zero confidence.

5 EXPERIMENTS AND RESULTS

5.1 DATASET VALIDATION AND ANALYSIS

We have collected a set of 3333 riddles from the Internet (puzzle websites). Each riddle has 4 images and a groundtruth answer associated with it. To make it more challenging to computer systems, we include both photographic and non-photographic images in the dataset.

To verify the groundtruth answers, we define the metrics: i) "correctness" - how correct and appropriate the answers are, and ii) "difficulty" - how difficult are the riddles. We conduct an Amazon Mechanical Turker (AMT)-based evaluation for dataset validation. We ask them to rate the correctness from 1-6[3]. The "difficulty" is rated from 1-7[4]. We provide the Turkers with examples to calibrate our evaluation. According to the Turkers, the mean correctness rating is 4.4 (with Standard Deviation

[3] 1: Completely gibberish, incorrect, 2: relates to one image, 3 and 4: connects two and three images respectively, 5: connects all 4 images, but could be a better answer, 6: connects all images and an appropriate answer.

[4] These gradings are adopted from VQA AMT instructions (Antol et al., 2015). 1: A toddler can solve it (ages:3-4), 2: A younger child can solve it (ages:5-8), 3: A older child can solve it (ages:9-12), 4: A teenager can solve it (ages:13-17), 5: An adult can solve it (ages:18+), 6: Only a Linguist (one who has above-average knowledge about English words and the language in general) can solve it, 7: No-one can solve it.

1.5). The "difficulty" ratings show the following distribution: toddler (0.27%), younger child (8.96%), older child (30.3%), teenager (36.7%), adult (19%), linguist (3.6%), no-one (0.64%). In short, the average age to answer the riddles is closer to **13-17yrs**. Also, few of these (4.2%) riddles seem to be incredibly hard. Interestingly, the average age perceived reported for the recently proposed VQA dataset (Antol et al., 2015) is **8.92 yrs**. Although, this experiment measures "the turkers' perception of the required age", one can conclude with statistical significance that the riddles are comparably harder.

5.2 SYSTEMS EVALUATION

The presented approach suggests the following hypotheses that requires empirical tests: I) the proposed approach (and their variants) attain reasonable accuracy in solving the riddles; II) the individual stages of the framework improves the final inference accuracy of the answers. In addition, we also experiment to observe the effect of using commercial classification methods like Clarifai against a published state-of-the-art Image Classification method.

5.2.1 Systems

We propose several variations of the proposed approach and compare them with simple vision-only baselines. We introduce an additional Bias-Correction stage after the Image Classification, which aims to re-weight the detected seeds using additional information from other images. The variations then are created to test the effects of varying the Bias-Correction stage and the effects of the individual stages of the framework on the final accuracy (hypothesis II). We also vary the initial Image Classification Methods (Clarifai, Deep Residual Network).

Bias-Correction: We experimented with two variations: i) greedy bias-correction and ii) no bias-correction. We follow the intuition that the re-weighting of the seeds of one image can be influenced by the others[5]. To this end, we develop the "GreedyUnRiddler" (**GUR**) approach. In this approach, we consider all of the images together to dictate the new weight of each seed. Take image $\mathcal{I}_k$ for example. To re-weight seeds in $\boldsymbol{S_k}$, we calculate the weights using the following equation: $\tilde{W}(s_k) = \frac{\sum_{j \in 1,..4} sim_{cosine}(V_{s_k,j}, V_j)}{4.0}$. V_j is vector of the weights assigned $\tilde{P}(\boldsymbol{S_j}|\mathcal{I}_j)$ i.e. confidence scores of each seed in the image. Each element of $V_{s_k,j}[i]$ is the ConceptNet-similarity score between the seed s_k and $s_{i,j}$ i.e. the i^{th} seed of the j^{th} image. The re-weighted seeds $(\boldsymbol{S_k}, \tilde{W}(\boldsymbol{S_k}))$ of an image are then passed through the rest of the pipeline to infer the final answers.

In the original pipeline ("UnRiddler",in short **UR**), we just normalize the weights of the seeds and pass on to the next stage. We experiment with another variation (called BiasedUnRiddler or **BUR**), the results of which are included in appendix, as **GUR** achieves the best results.

Effect of Stages: We observe the accuracy after each stage in the pipeline (**VB**: Up to Bias Correction, **RR**: Up to Rank and Retrieve stage, **All**: The entire Pipeline). For **VB**, we use the normalized weighted seeds, get the weighted centroid vector over the word2vec embeddings of the seeds for each image. Then we obtain the mean vector over these centroids. The top similar words from the word2vec vocabulary to this mean vector, constitutes the final answers. For **RR**, we get the mean vector over the top predicted targets for all images. Again, the most similar words from the word2vec vocabulary constitutes the answers.

Baseline (VQA+VB+UR): For the sake of completion, we experiment with a pre-trained Visual Question Answering system (from Lu et al. (2016)). For each image, we take top 20 answers for the question "What is the image about", and, then we follow the above procedure (**VB+UR**) to calculate the mean. We get the closest word using the mean vector, from the Word2vec vocabulary. We observe that, the detected words are primarily top frequent answers and do not contain any specific information. Therefore, subsequent stages hardly improve the results. We provide one detailed example in appendix.

Baseline (Clarifai+VB+UR and ResNet+VB+UR): We create a strong baseline by directly going from seeds to target using word2vec-based similarities. We use the class-labels and the confidence scores predicted using the state-of-the-art classifiers. For each image, we calculate the weighted centroid of the word2vec embeddings of these labels and the mean of these centroids for the 4 images. For the automatic evaluation we use top K (10) similar words and for human evaluation, we use the most similar word to this vector, from the word2vec vocabulary. The Baseline performances are listed in Table 3.

Human Baseline: In an independent AMT study, we ask the turkers to answer each riddle without any hint towards the answer. We ask them to input maximum 5 words (comma-separated) that can connect all four of the images. In cases, where the riddles are difficult we instruct them to find words that connect at least three images. These answers constitute our human baseline.

5.2.2 Experiment I: Automatic Evaluation

We evaluate the performance of the proposed approach on the Image Riddles dataset using both automatic and

[5] A person often skims through all the images at one go and will try to come up with the aspects that needs more attention.

Amazon Mechanical Turker (AMT)-based evaluations. An answer to a riddle may have several semantically similar answers. Hence, as evaluation metrics, we use both word2vec and WordNet-based similarity measures. For each riddle, we calculate the maximum similarity between the groundtruth with the top 10 detections, and report the average of such maximum similarities in percentage form: $S = \frac{1}{n}\sum_{i=1}^{n} \max_{1 \le l \le 10} sim(GT_i, T_l)$. To calculate phrase similarities, i) we use `n_similarity` method of the `gensim.models.word2vec` package; or, ii) average of WordNet-based word pair similarities that is calculated as a product of `length` (of the shortest path between sysnsets of the words), and `depth` (the depth of the subsumer in the hierarchical semantic net) (Li et al., 2006) [6].

Number of Targets: $\boldsymbol{\theta}_{\#t}$ (2500), ConceptNet-similarity Weight: $\boldsymbol{\theta}_{\alpha_1}$ (1),
word2vec-similarity weight: $\boldsymbol{\theta}_{\alpha_2}$ (4), Number of maximum similar Targets: $\boldsymbol{\theta}_{t\text{-}t}$ (1)
Seed-target similarity Threshold: $\boldsymbol{\theta}_{sim,psl1}$ (0.8),
Sum of confidence scores in Stage I: $\boldsymbol{\theta}_{sum1}$ (2)

Table 2: A List of parameters $\boldsymbol{\theta}$ used in the approach

			3.3k		2.8k	
			W2V	WN	W2V	WN
Human	-	-	74.6	68.9	74.56	67.8
VQA	VB	UR †	59.6	15.7	59.7	15.6
		GUR	62.59	17.7	62.5	17.7
Clarifai	VB	UR †	65	26.2	65.3	26.4
		GUR	65.3	26.2	65.36	26.2
	RR	UR	65.9	34.9	65.7	34.8
		GUR	65.9	36.6	65.73	36.4
	All	UR	68.5	**40.3**	68.57	**40.4***
		GUR	**68.8***	**40.3**	**68.7**	**40.4***
Resnet	VB	UR †	68.3	35	68	33.5
		GUR	66.8	33.1	66.4	32.6
	RR	UR	66.7	38.5	66.7	38.2
		GUR	66.3	38.1	66.2	37.6
	All	UR	68.53	39.9	68.2	**40.2**
		GUR	68.2	39.5	68.2	39.6

Table 3: Accuracy (in percentage) on the Image Riddle Dataset. Pipeline variants (VB, RR and All) are combined with Bias-Correction stage variants (GUR, UR). We show both word2vec and WordNet-based (WN) accuracies. (*- Best, † - Baselines).

To select the parameters in the parameter vector $\boldsymbol{\theta}$, We employed a random search on the parameter-space over first 500 riddles over 500 combinations. The final set of parameters used and their values are tabulated in Table 2.

The accuracies after different stages of the pipeline (VB, RR and All) combined with variations of the initial Bias-Correction stage (GUR and UR), are listed in Table 3[7]. We provide our experimental results on this 3333 riddles and 2833 riddles (barring 500 riddles as validation set for the parameter search).

5.2.3 Experiment II: Human Evaluation

We conduct an AMT-based comparative evaluation of the results of the proposed approach (GUR+All using Clarifai) and two vision-only baselines. We define two metrics: i) "correctness" and ii) "intelligence". Turkers are presented with the instructions: *We have three separate robots that attempted to answer this riddle. You have to rate the answer based on the correctness and the degree of intelligence (explainability).* The correctness is defined as before. In addition, turkers are asked to rate intelligence in a scale of 1-4[8]. Figure 3 plots the percentage of total riddles per each value of correctness and intelligence. In these histograms plots, we expect an increase in the rightmost buckets for the more "correct" and "intelligent" systems.

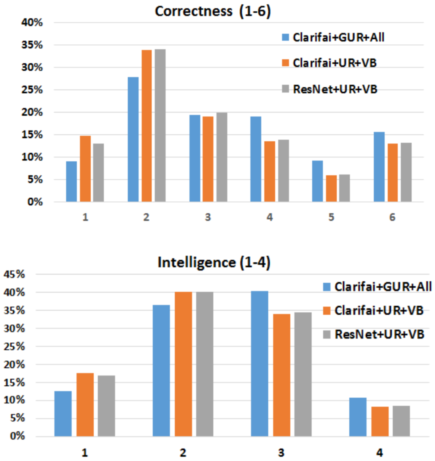

Figure 3: AMT Results of Thẹ Clarifai+GUR+All (our), Clarifai+UR+VB (baseline 1) and ResNet+UR+VB (baseline 2) approaches. Correctness Means are: 2.6 ± 1.4, 2.4 ± 1.45, 2.3 ± 1.4. For Intelligence: 2.2 ± 0.87, 2 ± 0.87, 1.8 ± 0.8

5.2.4 Analysis

Experiment I shows that the GUR variant (**Clarifai+GUR+All** in Table 3) achieves the best results in terms of word2vec-based accuracy. The WordNet-based metric gives clear evidence of improvement by the stages of our pipeline (a sharp **14**% increase over Clarifai and **6**% increase over ResNet baselines). Improvement from the final reasoning stage is also evident from the result. The increase in accuracy after reasoning shows how knowledge helped in decreasing overall uncertainty in perception. Similar trend is reflected in the AMT-based evaluations (Figure 3).

[6] The groundtruth is a single word. Code: bit.ly/2gqmnwEe.

[7] For ablation study on varying top K, check appendix.

[8] {1: Not, 2: Moderately, 3: Normal, 4: Very} intelligent

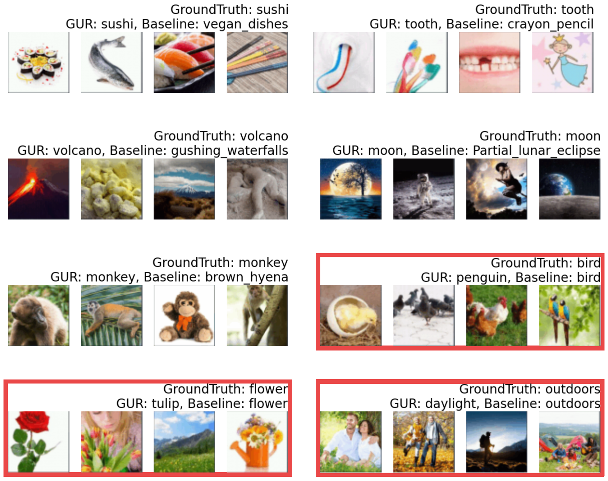
GroundTruth: sushi
GUR: sushi, Baseline: vegan_dishes
GroundTruth: tooth
GUR: tooth, Baseline: crayon_pencil
GroundTruth: volcano
GUR: volcano, Baseline: gushing_waterfalls
GroundTruth: moon
GUR: moon, Baseline: Partial_lunar_eclipse
GroundTruth: monkey
GUR: monkey, Baseline: brown_hyena
GroundTruth: bird
GUR: penguin, Baseline: bird
GroundTruth: flower
GUR: tulip, Baseline: flower
GroundTruth: outdoors
GUR: daylight, Baseline: outdoors

Figure 4: Positive and Negative (in red) results of the "GUR" approach (**Clarifai+GUR+All**) on some of the riddles. The groundtruth labels, closest label among top 10 from GUR and the Clarifai+VB+UR baseline are provided for all images. For more results, check Appendix.

Our system has increased the percentage of puzzles for the rightmost bins i.e. produces more "correct" and "intelligent" answers for more number of puzzles. The word2vec-based accuracy puts the performance of ResNet baseline close to that of the GUR variant. However, as evident from the WordNet-based metric and the AMT evaluation of the correctness (Figure 3), the GUR variant clearly predicts more meaningful answers than the ResNet baseline. Experiment II also includes what the turkers think about the intelligence of the systems that tried to solve the puzzles. This also puts the GUR variant at the top. The above two experiments empirically show that our approach achieves a reasonable accuracy in solving the riddles (Hypothesis I). In table 3, we observe how the accuracy varies after each stage of the pipeline (hypothesis II). The table shows a jump in the (WN) accuracy after the RR stage, which leads us to believe the primary improvement of our approach is attributed to the Probabilistic Reasoning model. We also provide our detailed results for the "GUR" approach using a few riddles in Figure 4.

Difficulty of Riddles: From our AMT study (**Human** baseline), we observe that the riddles are quite difficult for (untrained) human mechanical turkers. There are around 500 riddles which were labeled as "blank", another 500 riddles were labeled as "not found". Lastly, 457 riddles (391 with wordnet similarity higher than 0.9 and 66 higher than 0.8) were predicted perfectly, which leads us to believe that these easy riddles mostly show visual similarities (object-level) whereas others mostly show conceptual similarity.

Running Time: Our implementation of PSL solves each riddle in nearly $20s$ in an Intel core i7 2.0 GHz processor, with 4 parallel threads. Solving each riddle boils down to solving 5 optimization problems (1 for each image and 1 joint). This eventually means our engine takes nearly 4 sec. to solve an inference problem with approximately 20×2500 i.e. 50k rules.

Reason to use a Probabilistic Logic: We stated our reasons for choosing PSL over other available Probabilistic Logics. However, the simplicity of the used rules can leave the reader wondering about the reason for choosing a complex probabilistic logic in the first place. Each riddle requires an answer which is "logically" connected to each image. To show such logical connection, we need ontological knowledge graphs such as ConceptNet which shows connections between the answer and words detected from the images. To integrate ConceptNet's knowledge seamlessly into the reasoning mechanism, we use a probabilistic logic such as PSL.

6 CONCLUSION

In this work, we presented a Probabilistic Reasoning based approach that uses background knowledge to solve a new class of image puzzles, called "Image Riddles". We have collected over 3k such riddles. Crowd-sourced evaluation of the dataset demonstrates the validity of the annotations and the nature of the difficulty of the riddles. We empirically show that our approach improves on vision-only baselines and provides a stronger baseline for future attempts. The task of "Image Riddles" is equivalent to conventional IQ test questions such as analogy solving, sequence filling; which are often used to test human intelligence. This task of "Image Riddles" is also in line with the current trend of VQA datasets which require visual recognition and reasoning capabilities. However, it focuses more on the combination of both vision and reasoning capabilities. In addition to the task, the proposed approach introduces a novel inference model to infer related words (from a large vocabulary) given class labels (from a smaller set), using semantic knowledge of words. This method is general in terms of its applications. Systems such as (Wu et al., 2016), which use a collection of high-level concepts to boost VQA performance; can benefit from this approach.

7 ACKKNOWLEDGMENTS

The support of the National Science Foundation under the Robust Intelligence Program (1816039 and 1750082) is gratefully acknowledged.

References

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *International Conference on Computer Vision (ICCV)*, 2015. 2, 6, 7

Stephen Bach, Bert Huang, Ben London, and Lise Getoor. Hinge-loss markov random fields: Convex inference for structured prediction. *arXiv preprint arXiv:1309.6813*, 2013. 2, 3

David M. Blei. Probabilistic topic models. *Commun. ACM*, 55(4):77–84, April 2012. ISSN 0001-0782. doi: 10.1145/2133806.2133826. URL http://doi.acm.org/10.1145/2133806.2133826. 2

Clement Farabet, Camille Couprie, Laurent Najman, and Yann LeCun. Learning hierarchical features for scene labeling. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(8):1915–1929, 2013. 4

Haoyuan Gao, Junhua Mao, Jie Zhou, Zhiheng Huang, Lei Wang, and Wei Xu. Are you talking to a machine? dataset and methods for multilingual image question answering. In *NIPS*, 2015. 2

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jagannath Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 580–587. IEEE, 2014. 4

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90. URL http://dx.doi.org/10.1109/CVPR.2016.90. 2, 4

Derek Hoiem, Yodsawalai Chodpathumwan, and Qieyun Dai. Diagnosing error in object detectors. In *European conference on computer vision*, pages 340–353. Springer, 2012. 5

Angelika Kimmig, Stephen Bach, Matthias Broecheler, Bert Huang, and Lise Getoor. A short introduction to probabilistic soft logic. In *Proceedings of the NIPS Workshop on Probabilistic Programming: Foundations and Applications*, pages 1–4, 2012. 2, 3

GJ Klir and B Yuan. Fuzzy sets and fuzzy logic: theory and applications. 1995. 3

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 4

Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, pages 1–101, 2016. 1

Hugo Larochelle, Dumitru Erhan, Yoshua Bengio, Universit De Montral, and Montral Qubec. Zero-data learning of new tasks. In *In AAAI*, 2008. 2

Yuhua Li, David McLean, Zuhair A Bandar, James D O'shea, and Keeley Crockett. Sentence similarity based on semantic nets and corpus statistics. *IEEE transactions on knowledge and data engineering*, 18 (8):1138–1150, 2006. 8

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014*, pages 740–755. Springer, 2014. 2

H. Liu and P. Singh. Conceptnet - a practical commonsense reasoning tool-kit. *BT Technology Journal*, 22 (4):211–226, October 2004. ISSN 1358-3948. 2

Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Hierarchical question-image co-attention for visual question answering. In *Advances In Neural Information Processing Systems*, pages 289–297, 2016. 7

Lin Ma, Zhengdong Lu, and Hang Li. Learning to answer questions from image using convolutional neural network. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 3567–3573, 2016. URL http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/11745. 2

Mateusz Malinowski, Marcus Rohrbach, and Mario Fritz. Ask your neurons: A neural-based approach to answering questions about images. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 1–9, 2015. 2

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. 3

T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, and J. Welling. Never-ending learning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI-15)*, 2015. 3

Happy Mittal, Shubhankar Suman Singh, Vibhav Gogate, and Parag Singla. Fine grained weight learning in markov logic networks. 2015. 3

Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine learning*, 62(1-2):107–136, 2006. 3

Gaurav Sood. *clarifai: R Client for the Clarifai API*, 2015. R package version 0.2. 2, 4

Robert Speer and Catherine Havasi. Representing general relational knowledge in conceptnet 5. 2012. 3

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: A core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 697–706, New York, NY, USA, 2007. ACM. 3

Qi Wu, Chunhua Shen, Lingqiao Liu, Anthony Dick, and Anton van den Hengel. What value do explicit high level concepts have in vision to language problems? In *CVPR*, June 2016. 9