Building a Large-scale Multimodal Knowledge Base for Visual Question Answering

Yuke Zhu† Ce Zhang†‡ Christopher Ré† Li Fei-Fei†
†Computer Science Department, Stanford University
‡Computer Sciences Department, University of Wisconsin-Madison
{yukez,czhang,chrismre,feifeili}@cs.stanford.edu

Abstract

The complexity of the visual world creates significant challenges for comprehensive visual understanding. In spite of recent successes in visual recognition, a lack of common sense knowledge and the insufficiencies of joint reasoning still leave a large gap between computer vision models and their human counterparts for higher-level tasks such as question answering (QA). In this work, we build a multimodal knowledge base (KB) incorporating visual, textual and structured data, as well as their diverse relations for visual QA. Such a general-purpose approach presents major challenges in terms of scalability and quality. Our approach is to cast large-scale MRFs into a KB representation, and introduce a scalable knowledge base construction system by leveraging database techniques. We unify several visual QA tasks in a principled query language. Our KB achieves competitive results compared to purpose-built models, while exhibiting greater flexibility in visual QA.

1. Introduction

We have seen the recent and tremendous progress in perceptual tasks such as image classification and object detection [11, 15, 30, 37]. One next step forward is to focus on deeper understanding and visual reasoning. A number of new lines of work have emerged recently towards this goal. One is to cast the deeper understanding task into a sentence or story-generation task, such as image captioning trained with NLP inputs [7, 14, 17]. Another is to understand and recognize the image contents with more details or interconnected context, such as image recognition based on context [4, 24, 36] and joint learning and recognition of physical properties, functions, and appearance [34, 39, 42].

In this paper, we focus on a related but distinct task, which is vision-based reasoning by question answering (QA), especially with large-scale real-world data. Consider an iconic image from WWII in Fig. 1. While it is useful to label the picture with objects and scene classes, or even with a short descriptive sentence, it is also valuable to build a framework that is capable of answering more fluid questions about the picture. These questions can be based both on its pixel contents (e.g., what is the sailor doing?) as well as the related meta- and textual data. QA is a natural form of communication between humans, and is becoming important between humans and machines. Recently, Geman et al. [9] and Malinowski and Fritz [23] have also suggested that QA can potentially serve as an evaluation task for visual understanding beyond classification [22, 23, 34].

There have been well-known successes in text-based QA applications such as IBM’s Watson [8] and Google’s search engine [6]. But despite the ubiquitous presence of pictures and videos, QA based on images and other multimodal data is still a new area of exploration. While Geman et al. [9] and Malinowski and Fritz [23] focus on proposing a visual QA test for object detection problems, Lin and Parikh [22] have looked into richer reasoning of image descriptions using cartoon data. The most relevant work to ours is by Zhu et al. [42] on reasoning about object affordances. But all of these early work use toy-sized datasets of a small number of
images (on the order of one thousand images) and limited textual data (e.g., user- or researcher-generated phrases).

In this paper, we aim to build a scalable framework for visual QA tasks using multimodal data from the real world. In contrast to previous work [9, 22, 23] where questions come from a fixed pool and answers are restricted to text, our system aims to respond in the form of both text and images for a wider range of questions. Central to our model is the ability to perform joint learning and inference on a large amount of multimodal data, such as images, geolocations, time, textual labels, and other structured information, and to be able to perform various types of tasks with one unified query language. To do this, we use a Markov Random Field (MRF) representation. While MRFs have been widely used in a variety of vision tasks [4, 19, 20, 33], applying them to a large scale QA framework means that we need to overcome two challenges mentioned in the following two paragraphs.

One crucial property of a QA system is to handle large-scale multimodal data. There is a long history of influential work from the database and web communities, joined later by the NLP community, in using knowledge-base formulation for QA tasks [1, 2, 6]. In this paper, unless otherwise specified, we adopt the language of these communities and use the term Knowledge Base (KB) to refer to our MRF-based system. Previous visual KB works [3, 5, 42] use a standard representation [10, 29] from the NLP and database communities. This representation fails to incorporate continuous visual features in a probabilistic framework, which hinders us from expressing richer multimodal data. Our contribution is to cast the MRF model into a KB representation to accommodate a mixture of discrete and continuous variables in a joint probability model. Such a multimodal KB is the first attempt and a crucial step towards the goal of open-domain visual QA.

The second important challenge for a QA system is to handle large-scale learning and inference. As we will show in Sec. 4.2, even a modest data source of tens of thousands of images and a lexicon of hundreds of labels yields millions of model parameters. Traditional KB learning and inference methods (e.g., MLNs [29]) that rely on Boolean satisfiability cannot handle such large-scale multimodal data. Sampling methods, such as Gibbs, have the advantage of being able to handle both discrete and continuous variables. However, they typically take a long time to converge on large models with millions of parameters. Our contribution is to unite the recent advances in high-speed sampling [38] and first-order methods [26] with database techniques to scale up learning and inference. We are able to build a KB that is four orders of magnitude larger than previous work [42], while using only half of the training time. The resulting system has tenable scalability compared to previous KB systems used for large-scale NLP QA tasks [29, 41], and is also capable of learning from multimodal data.

We demonstrate the efficiency and effectiveness of our system by first learning a KB from our data sources [28, 31, 35] in Sec. 3.2. We then formalize visual QA tasks as evaluating marginal probabilities of queries in our joint probability model [32] in Sec. 5. We show the system’s flexibility in answering diverse types of questions in Sec. 6.1. We then perform a quantitative evaluation on two restricted QA tasks using the SUN dataset [35] in Sec. 6.2 and Sec. 6.3. Our experimental results illustrate that the general-purpose KB approach can achieve competitive performance with purpose-built models. Furthermore, it can be easily extended to a more general QA task.

2. Previous Work

Joint Models in Vision A series of context models have leveraged MRFs in various vision tasks, such as image segmentation [20, 24], object recognition [4, 19], object detection [33], pose and activity recognition [36] and other recognition tasks [13, 27]. Similarly, the family of And-Or graph models [34, 39] focus on parsing images and videos into a hierarchical structure. In this work, we use an MRF representation for joint learning and inference of our data, casting MRF models into modern KB systems.

Knowledge Bases Most knowledge base work in the NLP and database communities focuses on organizing and retrieving only textual information in a structured representation [2, 8, 21, 41]. Although a few large-scale KBs [2, 6] have made attempts to incorporate visual information, they simply cache the visual contents and link them to text via hyperlinks. In the vision community, Zitnick et al. [43] learned diverse relations from cartoon images based on occurrence and relative position. Chen et al. [3], Divvala et al. [5] and Zhu et al. [42] have recently proposed KB-based frameworks for visual recognition tasks. However, they all represent visual data in discrete variables [42] and binary relations [3, 5]. PhotoRecall [18] proposed a pre-defined knowledge structure to retrieve photos from text queries. In contrast, our system allows us to define new KB structures and offers the flexibility of answering richer types of queries.

Question Answering Knowledge bases play an integral role in real-world natural language QA systems [6, 8]. In the domain of visual QA, Malinowski and Fritz [23] proposed a multi-world approach to answering factual queries of scene images. Tu et al. [34] built a query answering system based on a joint parse graph from text and videos. Lin and Parikh [22] leveraged visual common sense learned from cartoons to solve two textual tasks. These preliminary works tackle restricted question answering tasks by purpose-built models. Our work proposes a KB framework towards a general, open-domain visual QA system.
3. A Joint Probability Model: Casting a Large-Scale MRF into a KB Representation

Our first task is to learn a knowledge base given a large amount of multimodal information, such as images, metadata, textual labels, and structured labels. As mentioned above, we would like our KB to possess a coherent probabilistic representation of both discrete and continuous variables, as well as a principled learning and inference method that is capable of large-scale computation. We address the first property in this section, and the second in Sec. 4.

3.1. Knowledge Base Representation

The knowledge base (KB) representation is an MRF model, where we represent the visual features as continuous variables, and textual and structured data as discrete variables. The structure of the MRF model captures the relations among these variables. This model provides an umbrella framework for visual QA, where we formalize question answering as evaluating corresponding marginals from the joint distribution. In comparison to MLNs used in previous KB work [29, 42], this representation is more generic, allowing us to accommodate continuous random variables and real-valued factors. In this representation, a KB can be intuitively thought of as a graph of nodes connected by edges as in Fig. 2, where the nodes are called “entities” and the edges are called “relations”.

In practice, we use factor graphs [16], a bipartite graph equivalence of an MRF. Factor graphs provide a simple graphical interpretation of the MRF model, which results in ease of implementation for large-scale inference later. Let $I$ be a possible world where every variable takes a certain value. A factor graph defines a joint probability distribution over all possible worlds. Let a factor $f$ be a real-valued function of variables. We define the probability of a possible world $I$ to be proportional to a log-linear combination of factors. We describe all the factors used in our KB in Sec. 3.2. Formally, given the set of all factors $F$, we define the partition function $Z$ of a possible world $I$ as

$$Z[I] = \prod_{i=1}^{|F|} \exp(w_i f_i) \quad (1)$$

where $w_i$ is the weight of the $i$-th factor $f_i$, and the probability of a possible world can be written as

$$\Pr[I; w] = \frac{Z[I]}{\sum_{I' \in \mathcal{I}} Z[I']} \quad (2)$$

where $\mathcal{I}$ is the set of all possible worlds, and $w$ corresponds to the factor weights that parameterize the joint distribution. In Fig. 2, each node corresponds to a random variable; and a relation between variables corresponds to a factor $f \in F$.

Figure 2: A graphical illustration of a visual knowledge base (KB). A visual KB contains both visual entities (e.g., scene images) and textual entities (e.g., semantic labels) interconnected by various types of edges characterizing their assorted relations. The colors of the nodes (edges) indicate different node (edge) types.

Each training image corresponds to a possible world $I$ in $\mathcal{I}_E$. Our learning objective is to find the optimal weight $w^* = \arg \min_w \sum_{I \in \mathcal{I}_E} \log \Pr[I; w]$. We discuss about how the gradient is computed with approximate inference in the supplementary material. We show in Sec. 4 that our system automatically creates a factor graph and performs scalable learning for knowledge base construction.

3.2. Data Sources for the KB

We now describe the entities and relations of our KB representation, and the data sources that we will use to populate the KB. For our purposes, SUN [35] is a particularly useful dataset because of a) its diverse set of images, and b) the availability of a number of category and attribute labels associated with the dataset.

Entities can be thought of as descriptors of the images. In the factor graph depicted in Fig. 2, they are the nodes (variables) of the graph.

Image feature nodes – are represented by their 4096-dimensional activations from the last fully-connected layer in a convolutional network [37]. These are continuous variables. In total, there are 59,709 images from the SUN dataset [35], where half are used for building the KB, and half are used for evaluation.

Image category labels – indicate scene classes. In our experiments, we use 15 basic-level categories (e.g., workplace and transportation), and 298 fine-grained level categories (e.g., grotto and swamp) from SUN [35].

Attribute labels – characterize visual properties (e.g., material, layouts, lighting, etc.) of a scene. The SUN Attribute Dataset [28] provides 102 attribute labels (e.g., glossy and warm).

Affordance labels – describe the functional properties of
a scene, i.e. the actions that one can perform in a scene. We used a lexicon of 227 affordances (actions). We conducted a large-scale online experiment to annotate the possibilities of the 227 actions for each scene category. Six images with their augmented annotations are shown in Fig. 3. We provide an exhaustive list of affordances in the supplementary material.

Relations link entities (variables) to each other, as depicted by the squares on the edges in Fig. 2. The weights learned for the edges (factors) indicate the strength of the relations. We introduce three types of relations in our model.

- **Image - label** – maps image features to semantic labels.
- **Intra-correlations** – capture co-occurrence between the pairwise attribute (affordance) labels.
- **Inter-correlations** – characterize correlations between two different types of labels (category-affordance, affordance-attribute, category-attribute and relations between categories from different levels).

Based on the representation described above, each image is associated with hundreds of attribute and affordance labels. Together, this amounts to a KB of millions of entities. Table 1 summarizes some of the basic statistics of the KB that will be learned. Comparing with Zhu et al. [42], this is more than a hundred times larger in terms of the number of entities and relations.

The large size of our dataset presents a significant challenge of scalability. In theory, an MRF can be arbitrarily large. However, its scalability is often subject to the inefficiency of learning and inference. In addition, it is prohibitive to handcraft such a large-scale model from scratch. We, therefore, need a principled and scalable system for constructing the visual KB.

---

1from the American Time Use Survey (ATUS) [31] sponsored by the Bureau of Labor Statistics, which catalogs the actions in daily lives and represents United States census data

2Our knowledge base construction system and all the data sources will be made available to the public.

3We provide the details of the declarative language and the complete list of rules in the supplementary material.
Figure 4: An overview of the knowledge base construction pipeline. We first process the images and text, converting them into a structured representation. We write human-readable rules to define the KB structure. The system automatically creates a factor graph by parsing the rules. We then adopt a scalable Gibbs sampler to learn the weights in the factor graph.

Figure 5: Efficiency of the knowledge base construction system. The curve is plotted in log-log scale, where the x-axis is the number of nodes in the factor graph, and the y-axis is the runtime to construct the knowledge base.

4.2. Learning Efficiency

The three steps (described in Sec. 4.1) together contribute to the high scalability of our KBC system. We now quantitatively analyze our system’s learning efficiency.\(^4\) We compare the scale of our KB with recent work [42] in building visual knowledge bases with MLNs [29] in Table 2.\(^5\) Our KB is four orders of magnitude larger in terms of the number of variables (in the grounded factor graph) and three orders of magnitude larger in terms of model parameters. At the same time, half of the learning time is required. Fig. 5 further demonstrates that the learning time grows steadily as the KB size increases. The end-to-end construction finishes in 5.2 hours on the whole dataset (Sec. 3.2), indicating the potential to build larger-scale KBs in the future.

\(^4\)The knowledge base construction is conducted on a Non-Uniform Memory Access (NUMA) machine [38] with four NUMA nodes. Each has 12 physical cores and 24 logical cores, with Intel Xeon CPU@2.40GHz and 1TB main memory.

\(^5\)Their KB construction [42] was done with Alchemy, an off-the-shelf MLN library [29] that cannot take advantage of the parallel system.

Table 2: Statistics of the Visual KB Systems

| variables | parameters | runtime |
|-----------|------------|---------|
| Zhu et al. [42] | 3.15 × 10^4 | 5.06 × 10^4 | 10 hr |
| Ours | 5.76 × 10^8 | 4.19 × 10^6 | 5.2 hr |

5. Visual QA Setup

As we have mentioned in the introduction, one advantage of using a KB system is its ability to handle QA tasks that are open-domain and multimodal. From a user’s perspective, the input to this system is a natural language question along with a set of one or more images. Similarly, the output (i.e. the answer) is a mixture of image(s) and text.

In practice, the space of possible QA tasks is huge (or even infinite). It is impossible to map each natural language question to the corresponding inference task in an ad-hoc manner. One solution is to reformulate the questions in a formal language [1], such as a probabilistic query language based on conjunctive queries [32]. This language allows us to express KB queries and to compute a ranked list of answers based on their marginal probabilities.

We briefly describe how this works by an example query that retrieves images of a sunny beach. This query is formed by a conjunction of two predicates (Boolean-valued functions) of sceneCategory and hasAttribute:

\[
\text{sceneCategory}(i, \text{beach}) \land \text{hasAttribute}(i, \text{sunny})
\]

Given such a query, our task is to find all possible images \(i\) where both predicates are true – i.e. image \(i\) comes from the scene category \(\text{beach}\) and has the attribute \(\text{sunny}\). Following this example, more complex queries can be expressed by joining several predicates together.\(^6\)

Let \(Q\) be a conjunctive query such as the one above. We compute a ranked list of answers (e.g., images of sunny...
beaches) based on their marginal probabilities. Formally, the marginal probability of a tuple \( t \) (a list of variable assignments) being an answer to \( Q \) is defined as:

\[
\Pr[t \in Q] = \sum_{I \in \mathcal{I}} I_{t \in Q(I)} \cdot \Pr[I]
\]

(3)

where \( \mathcal{I} \) and \( \Pr[I] \) are defined in Eq. (2), \( I \) is the indicator function, and \( Q(I) \) is the set of variable assignments of \( Q \) in the possible world \( I \) such that \( Q \) evaluates to true. We use the same Gibbs sampler as in Sec. 4.1 to estimate tuple marginals. The technical details of query evaluation can be found in the supplementary material. Each query evaluation produces a set of tuple-probability pairs \( \{(t_1, p_1), (t_2, p_2), \ldots\} \), where we retrieve the top answers by sorting the pairs based on their probabilities in a descending order.

6. Experiments

Now that we have learned a large KB from multimodal data sources, and have established a probabilistic query language that unifies different QA tasks, we can demonstrate how a KB can be useful in a number of different tasks. Ideally, we would have a standard open-domain visual QA benchmark, similar to the datasets [1] for NLP-based QA, which covers a broad range of visual tasks with multimodal answers including images and text. Such benchmark does not yet exist in the vision community. Hence, we perform several types of evaluations.

6.1. Answering Questions of Diverse Types

We start with a qualitative demonstration of using the KB to query a wide variety of questions based on image appearance, as well as metadata like geolocations, timestamps, and business information.\(^7\)

Fig. 6 provides a few query examples that depict the rich questions the system can handle. A user can ask the KB a question in natural language, such as “find me a modern looking mall near Fisherman’s Wharf.” While the photos of the malls are not part of the training data in Sec. 3.2, our system is capable of linking the appearances of the photos to other metadata, and is able to offer the names and locations of the shopping malls. Similarly in the second example “find me a place in Boston where I can play baseball”, our system predicts the affordances from the appearances of the photos, and combines them with geolocation information to retrieve a list of places for playing baseball. In Fig. 6, the answers are shown in a ranked list by their marginal probabilities. Without a joint probability model, previous work such as NEILL[3] and LEVAN [5] cannot produce such probabilistic outputs.

6.2. Single-Image Query Answering

While QA by KB is designed for answering a wide variety of questions, we can still evaluate how our system performs quantitatively in several standard visual recognition tasks, which is the focus of previous KB work [3, 5, 42]. Based on the KB we have learned from data sources such as SUN (see Sec 3.2), here we show two experiments for scene classification and affordance prediction. Both of

\(^7\)These metadata are either acquired from existing databases or automatically scraped online. Detailed descriptions of the experimental setups and the conjunctive queries (Sec. 5) for Fig. 6 are provided in the supplementary material.
these two tasks can be thought of as answering queries for a single image, where these queries can be expressed by a single predicate with the semantic labels taken as random variables – i.e. sceneCategory(img,c) and hasAffordance(img,a). We show that our system outperforms the state-of-the-art baseline methods trained for each of these tasks.

For both experiments, we use the data in Sec. 3.2 for training and an evaluation set of 29,781 images from the same 298 categories of SUN [35] for testing. We measure scene classification by mean accuracy (mAcc) over classes [40]. SUN [35] provides two ways of classification: basic-level (15 categories) and fine-grained (298 categories). Table 3 provides a summary of the results, comparing our full model (KB - Full) with a number of different settings and state-of-the-art models.

- **CNN Fine-tuned** We fine-tuned a CNN [37] on a subset of SUN397 dataset [35] of 107,754 images. We train $\ell_2$-logistic regression classifiers on the activations from the last fully-connected layer. We also use this as image features for all the other baselines.

- **Attribute-based model.** We predict the scene attributes and affordances from the CNN features, and use a binary vector of the predicted values as an intermediate feature. This is the strategy adopted by Zhu et al. [42] to discretize visual data.

- **Attributes + Features** We concatenate the predicted labels in Attribute-based model with CNN features as a combined representation.

- **KB - Affordance (Attributes)** A smaller KB learned without affordances (attributes).

- **KB - Full** Our full KB model defined in Sec. 3.2.

The Attributes + Features model (the third row in Table 3) outperforms the Attribute-based model (the second row in Table 3) by 11.7%, indicating the importance of modeling continuous features in the KB. The full model KB - Full achieves the state-of-the-art performance on both basic-level and fine-grained classes with more than 2% improvement over the CNN baseline.

| Method                      | Basic level | Fine-grained |
|-----------------------------|-------------|--------------|
| CNN Fine-tuned [37]         | 89.1        | 67.5         |
| Attribute-based model       | 88.0        | 57.9         |
| Attributes + Features       | 90.2        | 69.6         |
| KB - Affordances             | 90.0        | 69.3         |
| KB - Attributes              | 90.7        | 69.6         |
| KB - Full                    | **91.2**    | **69.8**     |

Table 3: Performance of Scene Classification (in mAcc)

**Fig. 7** offers some insight as to why a KB-based model performs well in a scene classification task. The class label is one of the many labels jointly inferred and predicted by the KB system, including attributes and affordances. So to predict an auditorium, attributes such as indoor lighting, enclosed area, and affordances such as taking class for personal interest can all help to reassure the prediction of an auditorium, and vice versa.

As mentioned in Sec. 3.2, we have collected annotations of 227 affordance classes for each of the 298 scene categories. We report the performance of affordance prediction by mean average precision (mAP) and mean F1 score (mF1) over the 227 affordance classes. The results are presented in Table 4. Here we compare our full KB model with the CNN Fine-tuned model [37], where we trained an $\ell_2$-logistic regression classifier on the CNN features for each of the 227 affordance classes. The KB - Full model outperforms the CNN baselines on both metrics.

Recall that the KB framework learns the weights of the relations between entities (e.g., scene classes, attributes and affordance, etc.) in a joint fashion. We can then examine the strength of these relations by looking at the factor weights of the underlying MRF. A large positive weight between two entities indicate a strong co-occurrence relation, whereas a large negative weight indicates a strong negative correlation. Fig. 8 provides examples of both the strongest and the weakest correlations between scene classes and attributes (Fig. 8(a)), as well as scene classes and affordances (Fig. 8(b)). For example, the KB has learned that the class beach has a strong co-occurrence relation with the attribute sand, and the class railroad track lacks correlation with the affordance teaching.

**6.3. Image Search by Text Queries**

Using the same model and framework, we can also query our KB for sets of images, instead of just one (Sec. 6.2), such as “find me images of a sunny beach where I can read books.” Here we use the same dataset as in Sec. 6.2.

We randomly generate 100 queries of a single label (scene category, affordance or attribute), and 100 queries of a pair of labels, each having at least 50 positive samples in the test set. Given a set of query labels, we aimed to retrieve the test images that are annotated with all the semantic labels in the set. We compare with two nearest neighbor baseline methods [12]. NNall ranks the test images based on the minimum Euclidean distance to any individual positive sample in the training set. NNmean ranks the images based on the distance to the centroids of the features of the

| Method                      | mF1  | mAP  |
|-----------------------------|------|------|
| CNN Fine-tuned [37]         | 81.6 | 74.2 |
| KB - Full                   | **82.6** | **75.7** |

Table 4: Performance of Scene Affordance Prediction
Figure 7: Sample prediction results by the full KB model. The ground-truth categories (in black) are shown in the first row. The first four images show examples of correct predictions from our KB model, and the last two show incorrect examples. Because our model jointly infers multiple labels of an image, we also show the predicted affordances (second row) in blue, and the predicted attributes (third row) in green.

7.29 beach sand
5.68 creek moist / damp
5.65 house shingles
-3.29 sun deck flowers
-3.69 apse indoor vinyl / linoleum
-3.86 gorge man-made
(a) Top weighted relations between categories and attributes

13.8 mountain snowy hunting
13.6 mountain participating in equestrian sports
12.5 orchard physical care of children
-0.94 call center medical services
-0.95 machine shop collecting as a hobby
-1.04 railroad track teaching
(b) Top weighted relations between categories and affordances

Figure 8: Examples of the strongest and the weakest relations in the learned KB. (a) Relations between scene classes (left column) and scene attributes (right column). (b) Relations between scene classes (left column) and scene affordances (right column). In both (a) and (b), the number at the beginning of each row indicates the actual factor weight in the underlying MRF. The more positive the number, the stronger the correlation. We show relations with the largest positive and negative weights in the KB. To be consistent with Fig. 7, we use the same color scheme for attributes and affordances.

Figure 9: (a) Performance variations of top k retrievals
We compare our method with two nearest neighbor baselines. In contrast to these two methods, the KB model maintains a steady performance on lower-ranked retrievals. (b) Top retrievals of example queries. We show top four retrievals from three sample queries (in bold) by our KB model. The green boxes indicate correct retrievals, and red ones indicate incorrect retrievals.

7. Conclusion
This paper presents a principled framework to learn and perform inference on a large-scale knowledge base (KB). Our contribution is to cast an MRF-based model into a KB by leveraging a number of recent advances in the database and NLP communities. By doing joint and contextual reasoning, our KB is capable of making predictions on a number of visual recognition tasks, on par with state-of-the-art models trained specifically for those tasks. In addition to these custom-trained classifiers, it is also interesting to explore these general-purpose, query-based knowledge representations as an initial attempt towards AI-complete open-domain visual QA tasks. Furthermore, this platform can be used to explore image-based reasoning. Towards these goals, future directions include a tighter integration between NLP and visual QA, and a more robust model for incorporating richer information.
References

[1] J. Berant et al. Semantic parsing on Freebase from question-answer pairs. EMNLP, 2013. 2, 5, 6
[2] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: A collaboratively created graph database for structuring human knowledge. SIGMOD, 2008. 2, 12
[3] X. Chen, A. Shrivastava, and A. Gupta. NEIL: Extracting visual knowledge from web data. ICCV, 2013. 2, 6
[4] C. Desai, D. Ramanan, and C. Fowlkes. Discriminative models for multi-class object layout. In ICCV, 2009. 1, 2
[5] S. Divvala, A. Farhadi, and C. Guestrin. Learning everything about anything: Webly-supervised visual concept learning. CVPR, 2014. 2, 6
[6] X. L. Dong et al. Knowledge Vault: A web-scale approach to probabilistic knowledge fusion. In KDD, 2014. 1, 2, 4
[7] A. Farhadi et al. Every picture tells a story: Generating sentences from images. ECCV, 2010. 1
[8] D. Ferrucci et al. Building Watson: An overview of the DeepQA project. AI Magazine, 2010. 1, 2, 4
[9] D. Geman, S. Geman, N. Hallonquist, and L. Younes. Visual Testing for computer vision systems. PNAS, 112(12):3618–3623, 2015. 1, 2
[10] L. Getoor, N. Friedman, D. Koller, and A. Pfeffer. Learning probabilistic relational models. In IJCAI, 1999. 2, 4
[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. arXiv preprint arXiv:1311.2524, 2013. 1
[12] K. Heller and Z. Ghahramani. A simple bayesian framework for content-based image retrieval. In CVPR, 2006. 7
[13] D. Hoiem, A. A. Efros, and M. Hebert. Putting objects in perspective. In CVPR, 2006. 2
[14] A. Krizhevsky and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. CVPR, 2015. 1
[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. NIPS, 2012. 1
[16] F. Kschischang, B. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. IEEE Trans. Inform. Theory, 2001. 3
[17] G. Kulkarni, V. Premraj, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg. Baby talk: Understanding and generating image descriptions. In CVPR, 2011. 1
[18] N. Kumar and S. Seitz. Photo recall: Using the internet to label your photos. In VSM Workshop at CVPR, 2014. 2
[19] S. Kumar and M. Hebert. Discriminative random fields: A discriminative framework for contextual interaction in classification. In ICCV, 2003. 2
[20] L. Ladicky, C. Russell, P. Kohli, and P. Torr. Graph cut based inference with co-occurrence statistics. ECCV, 2010. 2
[21] D. B. Lenat. Cyc: A large-scale investment in knowledge infrastructure. Commun. ACM, 1995. 2
[22] X. Lin and D. Parikh. Don’t just listen, use your imagination: Leveraging visual common sense for non-visual tasks. CVPR, 2015. 1, 2
[23] M. Malinowski and M. Fritz. A multi-world approach to question answering about real-world scenes based on uncertain input. In NIPS, 2014. 1, 2
[24] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille. The role of context for object detection and semantic segmentation in the wild. CVPR, 2014. 1, 2
[25] F. Niu, C. Ré, A. Doan, and J. Shavlik. Tuffy: Scaling up statistical inference in Markov logic networks using an RDBMS. Proc. VLDB Endow., 2011. 4
[26] F. Niu, B. Recht, C. Ré, and S. J. Wright. Hogwild!: A lock-free approach to parallelizing stochastic gradient descent. NIPS, 2011. 2, 4
[27] D. Parikh, C. L. Zitnick, and T. Chen. Exploring tiny images: The roles of appearance and contextual information for machine and human object recognition. PAMI, 2012. 2
[28] G. Patterson, C. Xu, H. Su, and J. Hays. The SUN attribute database: Beyond categories for deeper scene understanding. IJCV, 2014. 2, 3
[29] M. Richardson and P. Domingos. Markov logic networks. Machine learning, 2006. 2, 3, 4, 5
[30] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. IJCV, 2015. 1
[31] K. Shelley. Developing the american time use survey activity classification system. Monthly Labor Review, 2005. 2, 4, 13
[32] D. Succi, D. Olteanu, C. Ré, and C. Koch. Probabilistic Databases. Morgan-Claypool, 2011. 2, 5
[33] A. Torralba, K. Murphy, and W. Freeman. Contextual models for object detection using boosted random fields. In NIPS, 2005. 2
[34] K. Tu, M. Meng, M. W. Lee, T. E. Choe, and S.-C. Zhu. Joint video and text parsing for understanding events and answering queries. IEEE MultiMedia, 21(2):42–70, 2014. 1, 2
[35] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba. SUN database: Large-scale scene recognition from abbey to zoo. In CVPR, 2010. 2, 3, 7, 13
[36] B. Yao and L. Fei-Fei. Modeling mutual context of object and human pose in human-object interaction activities. In CVPR, 2010. 1, 2
[37] M. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In ECCV, 2014. 1, 3, 7, 10
[38] C. Zhang and C. Ré. DimmWitted: A study of main-memory statistical analytics. Proc. VLDB Endow., 2014. 2, 4, 5
[39] Y. Zhao and S.-C. Zhu. Scene parsing by integrating function, geometry and appearance models. CVPR, 2013. 1, 2
[40] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning Deep Features for Scene Recognition using Places Database. NIPS, 2014. 7
[41] J. Zhu, Z. Nie, X. Liu, B. Zhang, and J.-R. Wen. StatSnowball: a statistical approach to extracting entity relationships. In WWW, 2009. 2
[42] Y. Zhu, A. Fathi, and L. Fei-Fei. Reasoning about object affordances in a knowledge base representation. ECCV, 2014. 1, 2, 3, 4, 5, 6, 7
[43] C. L. Zitnick, D. Parikh, and L. Vanderwende. Learning the visual interpretation of sentences. In ICCV, 2013. 2
A. Scalable Knowledge Base Construction

There are three key steps to make the knowledge base construction (KBC) scalable: data processing, factor graph generation and high-performance learning. Sec. 4.1 provides an overview of the KBC process illustrating these three steps. Here we provide more detailed explanations of the database schema and the human-readable rules.

A.1. Database Schema

The first step (the first arrow in Fig. 4) is to process raw data into a structured representation. This representation enables us to perform structured queries (e.g. SQL) on the data. Each database table stores entities of a certain type. We build a database table for each of the four entity types in Sec. 3.2, where continuous variables (visual features) are stored as double precision numbers, and discrete variables (meta- and textual data) are stored as bigint in Greenplum database.8 We represent images by their activations extracted from a fine-tuned CNN [37]. Each dimension is one row in the database table. We provide the complete database schema in Fig. 10. The schema contains two types of tables: data tables contain the entities in Sec. 3.2 that are used to build the knowledge base (KB); metadata tables provide auxiliary information for the experiments and visualization.

| Table Name           | id       | sample_id | dimension | feat     | name     | attribute_id | label     |
|----------------------|----------|-----------|-----------|----------|----------|--------------|-----------|
| image features       | bigint   | bigint    | bigint    | double precision |         |              |           |
| scene categories     | bigint   | bigint    | bigint    | bigint   | bigint   | bigint       |           |
| scene affordances    | bigint   | bigint    | bigint    | bigint   | bigint   | bigint       |           |
| scene attribute names| bigint   | bigint    | bigint    | text     |          |              |           |
| scene affordance names| bigint  | bigint    | bigint    | text     |          |              |           |

8http://www.pivotal.io/big-data/pivotal-greenplum-database

Figure 10: Database schema for data storage. The table names (in bold), column names (left) and data types (right) are provided. The blue boxes denote data tables containing KB entities; and the green ones denote metadata tables. The id column is a unique identifier for each row, which is used to create the factor graph. The stars (*) indicate the distribution keys for parallel data processing.

We choose Greenplum as the underlying database sys-

tem due to its power in massive parallel data processing. sample_id in Fig. 10 is a unique identifier of each training sample. These identifiers are used as a distribution key in the database system, where the data is distributed across segments as per the distribution keys.

A.2. Human-readable Rules

To define the KB with ease, we develop a declarative language, which serves as a human-readable interface for specifying the KB structure. The syntax of the declarative language is an extension to first-order logic in order to accommodate continuous variables.

We introduced in Sec. 3.2 three types of relations. We define each type of relations by a group of rules, where each rule Rj is a set specified with first-order logic formulas. For example, to define the co-occurrence relations between sceneCategory and hasAffordance, we can simply write a rule like

\[ R = \{(i, w(s, a), 1) \mid \text{sceneCategory}(i, s) \land \text{hasAffordance}(i, a)\} \]

which specifies the strength of the co-occurrence relations between a scene category s and an affordance a. This rule indicates that an image i should have both sceneCategory (i, s) and hasAffordance (i, a) to be true with a confidence score of w(s, a). More generally, each rule Rj is a set in a given possible world I:

\[ I(R_j) = \{(\bar{x}, w(\bar{y}), f(\bar{z}))\} \quad (4) \]

where \(\bar{x}, \bar{y}, \bar{z}\) are sets of variable in the domain (the set of all possible values the variables can take), and \(w(\cdot), f(\cdot)\) are real-valued functions. Here \(f(\cdot)\) are essentially factors in the factor graph model (see Sec. 3.1), and \(w(\cdot)\) are the corresponding factor weights.

All three types of relations can be specified as rules written in this declarative language. Fig. 11 provides the complete list of rules that we have used to build the visual KB. To be more specific, we express image-label relations using two sets of rules corresponding to the linear terms and the bias terms as in logistic regression. For intra- and inter-correlations, we express them as conjunctions of two different predicates, which capture the co-occurrence between these two labels. In total, the proposed declarative language enables us to define the KB structure with eighteen first-order logic rules.

Our KBC system automatically parses these rules and creates a factor graph (the second arrow in Fig. 4). Now we have the structure of the factor graph model, the next step is to learn the model parameters (i.e., factor weights). We will talk about the details of learning and inference in the next section.
A.3. Learning and Inference

Given the structure of the factor graph model, the last step of KB construction (the third arrow in Fig. 4) is to obtain the factor weights. We use a sampling approach to learning these parameters, where inference is performed in the learning process. Hence, we start by introducing the mathematical details of the inference task first. Then we describe how learning is done with the inference method.

The inference task is to derive the marginal probabilities of a conjunctive query (see Sec. 5). This problem can be regarded as computing the expectation of a real function \( f : I \rightarrow \mathbb{R} \) given the probability distribution of possible worlds \( I \in \mathcal{I} \):

\[
E[f; w] = \sum_{I \in \mathcal{I}} \Pr[I; w] f(I)
\]

(5)

where \( \Pr[I; w] \) is the probability of a possible world \( I \) defined in Eq. (2), and \( \mathcal{I} \) is the set of all possible worlds. Computing the exact expectation in Eq. 5 is intractable in general factor graphs, which requires summing over a large (or even infinite) number of variable assignments. Gibbs sampling is a commonly used method for approximate inference.

The Gibbs sampling starts with an initial world \( I^{(0)} \). For each random variable \( v_k \) in the factor graph, we sample its new value \( v'_k \) from the conditional distribution \( p(v'_k | MB(v_k); w) \), where \( MB(v) \) is the Markov blanket of the variable \( v \) (i.e., the set of factors that are connected to the variable \( v \)). The sampler then moves to the next variable. After \( m \) rounds of iterations, we have sampled a set of possible worlds \( \Omega = \{ I^{(0)}, I^{(1)}, \ldots, I^{(m)} \} \). We thus approximate the expectations of a query \( q \) in Eq. (5) over \( \Omega \):

\[
\hat{E}[q] = \frac{1}{m} \sum_{i=1}^{m} q(I^{(i)})
\]

(6)

where \( q \) is a Boolean-valued function that computes the value of the conjunctive query \( q \) in a possible world. After sufficient iterations, the probability of a query can be approximated by the number of iterations in which it takes that value over the total number of iterations.

During learning, we optimize the factor weights using (approximate) stochastic gradient descent (SGD), where the gradient of a factor weight \( w_k \) is given by

\[
\nabla w_k = E[f_k(I') ; w | I' \in \mathcal{I}_E] - E[f_k(I'') ; w],
\]

(7)

where \( f_k \) is the corresponding factor of the weight \( w_k \), and \( \mathcal{I}_E \) is the set of all the possible worlds that are consistent with training data. Computing the exact gradient in Eq. (7) is intractable. We use contrastive divergence (similar to RBM training) as an estimator of the log-likelihood gradient:

\[
\nabla w_k \approx f_k(I') - f_k(I'') \quad I' \in \mathcal{I}_E,
\]

(8)

\[\text{image-label relations}\]

image features & scene category
\{(i,w(d),f) | sceneCategory(i,c) \land hasFeature(i,d,f)\}
\{(i,w(c),1) | sceneCategory(i,c)\}

image features & scene affordance
scene_affordance_and_scene_features
\{(i,w(a),f) | hasAffordance(i,a) \land hasFeature(i,d,f)\}
\{(i,w(a),1) | hasAffordance(i,a)\}

image features & scene attribute
\{(i,w(d),f) | hasAttribute(i,a) \land hasFeature(i,d,f)\}
\{(i,w(a),1) | hasAttribute(i,a)\}

INTRA-CORRELATIONS

affordance & affordance
\{((i,a1,a2), w(a1,a2), 1) | hasAffordance(i,a1) \land hasAffordance(i,a2)\}
\{((i,a1,a2), w(a1,a2), 1) | !hasAffordance(i,a1) \land hasAffordance(i,a2)\}
\{((i,a1,a2), w(a1,a2), 1) | hasAffordance(i,a1) \land !hasAffordance(i,a2)\}
\{((i,a1,a2), w(a1,a2), 1) | !hasAffordance(i,a1) \land !hasAffordance(i,a2)\}

attribute & attribute
\{((i,a1,a2), w(a1,a2), 1) | hasAttribute(i,a1) \land hasAttribute(i,a2)\}
\{((i,a1,a2), w(a1,a2), 1) | !hasAttribute(i,a1) \land hasAttribute(i,a2)\}
\{((i,a1,a2), w(a1,a2), 1) | hasAttribute(i,a1) \land !hasAttribute(i,a2)\}
\{((i,a1,a2), w(a1,a2), 1) | !hasAttribute(i,a1) \land !hasAttribute(i,a2)\}

INTER-CORRELATIONS

category & attribute
\{((i,c,a), w(a,c), 1) | sceneCategory(i,c) \land hasAttribute(i,a)\}
\{((i,c,a), w(a,c), 1) | sceneCategory(i,c) \land !hasAttribute(i,a)\}
\{((i,c,a), w(a,c), 1) | !sceneCategory(i,c) \land hasAttribute(i,a)\}
\{((i,c,a), w(a,c), 1) | !sceneCategory(i,c) \land !hasAttribute(i,a)\}

category & affordance
\{((i,c,a), w(a,c), 1) | sceneCategory(i,c) \land hasAffordance(i,a)\}
\{((i,c,a), w(a,c), 1) | sceneCategory(i,c) \land !hasAffordance(i,a)\}
\{((i,c,a), w(a,c), 1) | !sceneCategory(i,c) \land hasAffordance(i,a)\}
\{((i,c,a), w(a,c), 1) | !sceneCategory(i,c) \land !hasAffordance(i,a)\}

\[\text{Figure 11: The complete list of rules for the visual knowledge base construction. We build our visual knowledge base with the rules above. ! denotes negation and \land \land denotes conjunction. The formal semantics of the rules are described in Sec. A.2.}\]

where \( I' \) is a possible world sampled from the training data (evidence), and \( I'' \) is an evidence-free possible world sampled from the model parameterized by the current weights \( w \). We use \( \ell_2 \)-regularization on the factor weights.

B. Visual Question Answering

In Fig. 6, we have provided six question answering examples that illustrate the rich questions our QA system can handle. In order to answer these diverse types of questions, it requires a fusion of information from various sources. In practice, we aggregate information from online databases,
We augment our KB in Sec. 3.2 with a new set of geotagged images and several types of metadata. We briefly introduce the extra data sources that we used for this experiment in Sec. 6.1. We randomly sample from Flickr100M\(^9\) a pool of 20k images with geo-tags and timestamps. Besides these images, we incorporate additional information by either downloading from existing databases or crawling from the web.

1. We obtain a list of names and dates of 327 public holidays from Freebase\(^{10}\) [2] from the instances of /time/holiday\_category/holidays.

2. We scrape business information from Yelp.com and Hotels.com. We have crawled in total over sixteen thousand entries of business information, including 7k bars, 6k shopping centers and 3k hotels.

3. We download the daily temperature and weather data from National Climatic Data Center. Climate Data Online\(^11\) (CDO) provides free access to global historical weather and climate data.

4. We download the publicly available GeoNames geographical database\(^12\), which maps geolocations to over eight million place names.

All the information is stored in a structured format as database tables (Sec. A.1). We introduce new predicates (Boolean-valued functions) that enable us to query with these additional data. We omit the elaboration of the detailed definitions of these predicates, which can be easily inferred from the predicate names and the names of input variables.

Having defined the predicates, we can easily use the augmented KB to answer the questions in Fig. 6. We list the conjunctive queries for each of the six example queries in Fig. 12. \texttt{answer (·)} indicates the return variables (i.e. the answers) of the queries. We retrieve a ranked list of the answers by computing a marginal probability of the queries (see Sec. 5 and Sec. A.3).

Following this approach, we can express richer and more complex queries by joining different pieces of information with logical conjunctions. As we can see, the query language in Sec. 5 is capable of expressing a wide range of queries. Moreover, these queries can be answered in a principled manner. Given such a general-purpose framework,

---

\(^9\)http://yahoolabs.tumblr.com/post/89783581601/one-hundred-million-creative-commons-flickr-images
\(^10\)https://www.freebase.com
\(^11\)http://www.ncdc.noaa.gov/cdo-web/
\(^12\)http://www.geonames.org/
data becomes the key to extend our model’s power of answering real-world questions. We are interested in exploring more efficient and automatic ways to aggregate information from large-scale multimodal corpora for future work.

C. Affordance Annotations

We augment the SUN dataset [35] with additional annotations of scene affordances. We use a lexicon of 227 affordances (actions) from the American Time Use Survey (ATUS) [31] sponsored by the Bureau of Labor Statistics, which catalogs the actions in daily lives and represents United States census data. The original ATUS lexicon includes 428 specific activities organized into 17 major activity categories and 105 mid-level categories. We re-organize the categories by collapsing visually similar superordinate categories into one action. For instance, the superordinate-level category “traveling” was collapsed into a single category because being in transit to go to school should be visually indistinguishable from being in transit to go to the doctor. This results in 227 actions in total.

The lexicon covers a broad space of possible actions that could take place in scenes. We conducted a large-scale online experiment with over 400 AMT workers annotating the possibilities of the 227 actions for each of the 298 scene categories (Sec. 3.2). 10 votes are collected for each category-affordance pair. Positive (≥ 3 votes) and negative (≤ 2 votes) annotations are listed as evidence. These 227 affordances are listed in alphabetic order below:

A appliance repair & maintenance (self), architecture and engineering work, arts & crafts, arts & crafts with children, arts / design / entertainment / sports / media work, attending child’s events, attending meetings for personal interest, attending movies, attending museums, attending or hosting parties, attending religious services, attending school-related meetings & conferences, attending the performing arts

B banking, biking, boating, bowling, building & repairing furniture, building and grounds cleaning and maintenance work, business and financial operations work, buying / selling real estate

C camping, civic obligations, cleaning home exterior, collecting as a hobby, community and social work, comparison shopping, computer and mathematical work, computer use (not games), construction and extraction work

D dancing, doing aerobics, doing gymnastics, doing martial arts

E eating & drinking, education and library work, education-related administrative activities, email, exercising & playing with animals, exterior home repair & decoration, extracurricular club activities

F farming / fishing and forestry work, fencing, financial management, fishing, food & drink preparation, food preparation and serving work, food presentation

G gambling, golfing, grocery shopping

H health-related self care, healthcare work, helping adult, helping child with homework, hiking, hobbies, home heating / cooling, home security, home-schooling children, homework, household organization & planning, hunting

I in transit / traveling, income-generating hobbies & crafts, income-generating performance, income-generating rental property activity, income-generating selling activities, income-generating services, installation / maintenance and repair work, interior decoration & repair, interior home cleaning

J job interviewing, job search activities

K kitchen & food clean-up

L laundry, lawn / garden & plant care, legal work, listening to music (not radio), listening to radio, looking after adult, looking after children

M mailing, maintaining home pool / pond / hot tub, management / executive work, military work

N non-veterinary pet care

O obtaining licenses & paying fees, obtaining medical care for adult, obtaining medical care for child, office and administrative work, organizing & planning for adults, organizing & planning for children, out-of-home medical services

P participating in aquatic sports, participating in equestrian sports, participating in rodeo, personal care and service work, physical care of adults, physical care of children, picking up / dropping off adult, picking up / dropping off child, playing baseball, playing basketball, playing billiards, playing football, playing games, playing hockey, playing racquet sports, playing rugby, playing soccer, playing softball, playing sports with children, playing volleyball, playing with children (not sports), production work, protective services work, providing medical care to adult, providing medical care to child, purchasing food (not groceries), purchasing gasoline

R reading for personal interest, reading with children, relaxing, religious education, religious practices, rock climbing / caving, rollerblading / skateboarding, running

S sales work, school music activities, science work, security screening, sewing & repairing textiles, sexual activity, shopping (except food and gas), skiing / ice skating / snowboarding, sleeping, socializing, storing household items, student government

T taking class for degree or certification, taking class for personal interest, talking with children, telephone calls, tobacco use, transportation and material moving work, travel, using cardiovascular equipment

U using clothing repair & cleaning services, using home repair & construction services, using in-home medical services, using interior home cleaning services, using lawn & garden services, using legal services, using meal preparation services, using other financial services, using paid childcare services, using personal care services, using pet services, using police & fire services, using professional