KRISP: Integrating Implicit and Symbolic Knowledge for Open-Domain Knowledge-Based VQA

Kenneth Marino[1,2*] Xinlei Chen[2] Devi Parikh[2,3] Abhinav Gupta[1,2] Marcus Rohrbach[2]

[1]Carnegie Mellon University [2]Facebook AI Research [3]Georgia Tech

Abstract

One of the most challenging question types in VQA is when answering the question requires outside knowledge not present in the image. In this work we study open-domain knowledge, the setting when the knowledge required to answer a question is not given/annotated, neither at training nor test time. We tap into two types of knowledge representations and reasoning. First, implicit knowledge which can be learned effectively from unsupervised language pretraining and supervised training data with transformer-based models. Second, explicit, symbolic knowledge encoded in knowledge bases. Our approach combines both—exploiting the powerful implicit reasoning of transformer models for answer prediction, and integrating symbolic representations from a knowledge graph, while never losing their explicit semantics to an implicit embedding. We combine diverse sources of knowledge to cover the wide variety of knowledge needed to solve knowledge-based questions. We show our approach, KRISP (Knowledge Reasoning with Implicit and Symbolic rePresentations)*, significantly outperforms state-of-the-art on OK-VQA, the largest available dataset for open-domain knowledge-based VQA. We show with extensive ablations that while our model successfully exploits implicit knowledge reasoning, the symbolic answer module which explicitly connects the knowledge graph to the answer vocabulary is critical to the performance of our method and generalizes to rare answers.* [1]

1. Introduction

Consider the example shown in Fig. 1. To answer this question, we not only need to parse the question and understand the image but also use external knowledge. Early work in VQA focused on image and question parsing [2, 6, 23, 49, 50] assuming all required knowledge can be learned from the VQA training set. However, learning knowledge from image-question-answer triplets in the training data is not scalable and is liable to biases in the training data. We should exploit other external knowledge sources such as Wikipedia or knowledge graphs. The recent OK-VQA dataset [51] consists of these types of questions and allows us to study open-domain knowledge in VQA.

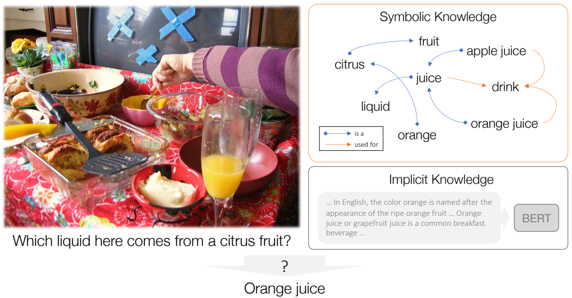

Which liquid here comes from a citrus fruit?
?
Orange juice
Symbolic Knowledge
fruit
apple juice
citrus
juice
drink
liquid
orange juice
orange
is a
used for
Implicit Knowledge
... In English, the color orange is named after the appearance of the ripe orange fruit ... Orange juice or grapefruit juice is a common breakfast beverage ...
BERT

Figure 1. An OK-VQA [51] example that requires external knowledge. Our KRISP model uses a symbolic knowledge graph as well as the implicit knowledge learned from large-scale BERT training to answer the question.

We can define two types of knowledge representation that can be useful for these types of questions: First we have implicit knowledge, knowledge which is embedded into some non-symbolic form such as the weights of a neural network derived from annotated data or large-scale unsupervised language training. Recently, transformer- and specifically BERT- [16] based multi-modal VQA models have been proposed [40, 46, 47], which incorporate large scale language pretraining, implicitly capturing language based, as well as multimodal knowledge. This type of knowledge can be quite useful, but we find this form of implicitly learned knowledge is not sufficient to answer many knowledge-based questions as we will show. Perhaps this is not surprising if one considers that many facts are rare such as "Thomas Newcomen invented the steam engine" and learning them with implicit representations might be less efficient while there are external sources and knowledge bases that state it explicitly.

The other type of knowledge typically studied is explicit or symbolic knowledge, often in the form of knowl-

[*] Work done during internship at Facebook

[1] Code and more are available at https://github.com/facebookresearch/krisp

edge graphs. Approaches that use this form of knowledge either take the symbolic knowledge and then embed-and-fuse them into a larger VQA model before answer prediction which no longer maintains the well-defined knowledge structures [51, 39], or by relying on a closed set of knowledge facts with strong annotation of source knowledge [54, 74, 77]. In the second case, the VQA dataset itself has ground truth "facts" associated with the question, so solving these questions often ends up being the problem of retrieving a fact from the closed set. In our method, we preserve the symbolic meaning of our knowledge from input until answer prediction. This allows us to use knowledge that is rare or is about rare entities as learning the reasoning logic with symbols is shared across all symbols. And unlike other work, we do not have a closed set or ground truth knowledge, so we must build a large diverse knowledge base for use by our model.

In this work, we develop an architecture, *KRISP (Knowledge Reasoning with Implicit and Symbolic rePresentations)*, to successfully combine the implicit and symbolic knowledge. Specifically, KRISP uses (i) a multi-modal BERT-pretrained transformer to process the question and image, and take advantage of the implicit knowledge in BERT, and (ii) a graph network to make use of symbolic knowledge bases. To cover the wide variety of knowledge required in OK-VQA, we draw on four very different knowledge sources to construct our knowledge graph: DBPedia [7], ConceptNet [44], VisualGenome [36] and hasPart KB [10]. This covers crowdsourced data, visual data, encyclopedic data, knowledge about everyday objects, knowledge about science and knowledge about specific people, places and events. Finally, our method preserves the symbolic meaning of the knowledge by making predictions based on the hidden state of individual nodes in the knowledge graph and using a late-fusion strategy to combine the implicit and symbolic parts of the model.

2. Related Work

Multimodal Vision and Language Modeling. Approaches for multimodal vision and language tasks have explored diverse set of fusion strategies such as bilinear models (*e.g.* [24, 33]) or self-attention (*e.g.* [25]). Many recent works have been inspired by the success of transformer [71] and BERT [16] models for natural language tasks and proposed transformer-based fusion between image and text [3, 15, 38, 40, 46, 69, 70, 84]. Similar to these works as part of our method we train a multimodal transformer with BERT-pretraining to import the implicit knowledge learned by BERT and learn any knowledge encoded in the training data and study it on knowledge VQA.

Another line of work has been extracting programs from the question for explicit reasoning with modules [5] or extracting symbols from the image to reason over them [82]. These works focus on reasoning about things explicitly in the image but do not integrate external knowledge.

Knowledge in Computer Vision. Knowledge has a long history in computer vision problems. Some of the earliest versions of this work was relating to attributes [19, 67] or knowledge mined from the web [63], often for zero- or few-shot learning problems [20, 37, 62], as well as for fine-grained classification [18]. The use of word embeddings from language has been extensive including in [22, 35, 45]. Class hierarchies such as WordNet [53] have often been used to aid in image recognition [85, 60]. Knowledge graphs have also found extensive use in visual classification and detection [52, 13], zero-shot classification [76] and image retrieval [31]. In our work we also rely on a knowledge graph to represent symbolic knowledge.

Knowledge-based VQA datasets. While open-ended VQA datasets (*e.g.* [6]) might require outside knowledge to answer some of its questions which cannot be learned from the dataset, there are a few datasets which focus specifically on knowledge based multi-modal reasoning. One is FVQA [74], where image-questions-answer triples are annotated with a fact-triple (*e.g.* "chair is furniture") from a fixed outside knowledge base, which allows deriving the answer. Specifically one of the two nodes (*i.e.* chair or furniture in this example) is the answer. A more recent and more challenging dataset is OK-VQA [51] which stands for *Open Knowledge VQA*, as the name suggests, focusing on knowledge which is not tied to a specific knowledge base. In this work we focus our evaluation on OK-VQA due to its relatively large number of knowledge-based questions, as well as its challenging and open-ended nature.

Symbolic Knowledge for VQA. Symbolic knowledge from knowledge bases is commonly represented as graphs/knowledge bases [39, 54, 55, 73, 74] or textual knowledge sources such as Wikipedia [51, 77]. We can separate these into two directions: where symbols are retained until prediction and where they are not. [54, 73, 74] retain the symbols until the answers, allowing good generalization capabilities but require annotations of the "correct" knowledge fact and are difficult to generalize to open knowledge VQA. For improved generalization to open-domain VQA, [26, 51, 39, 77] embed the symbolic knowledge to an implicit embedding loosing the semantics of the symbols, but therefore are able to easily integrate the embedding with standard VQA approaches. Similar to our work, the recent work [26] relies on a multimodal transformer model (pretrained VilBERT [46], however, similar to the other works it looses the semantics of the knowledge symbols when it integrates over them with an attention model. In contrast, our work shows how to take advantage of both the implicit and symbolic knowledge directions: We retain symbols until the end without the need of knowledge-fact annotations and integrate it with implicit knowledge and powerful reasoning abilities of multi-modal transformers.

Knowledge Bases & Knowledge in NLP. There have been many knowledge bases proposed for knowledge-based reasoning, both language-only and multi-modal [85, 14, 17, 65, 88, 87, 10, 53, 36]. In the NLP literature, there has been much work in question answering from knowledge sources [9, 81, 11] including for open-domain question answering [12, 75, 80, 79], and including mixed symbolic/implicit methods for question answering [48, 32].

3. The KRISP Model

In this section we introduce our model: *Knowledge Reasoning with Implicit and Symbolic rePresentations* (KRISP). An overview of our model can be seen in Fig. 3. We first introduce our transformer-based multi-modal implicit knowledge reasoning (Sec. 3.1), then discuss the symbolic knowledge sources and reasoning with symbols (Sec. 3.2), and then describe their integration in Sec. 3.3.

3.1. Reasoning with Implicit Knowledge

We want to incorporate implicit external knowledge as well as multi-modal knowledge which can be learned from training set in our model. Language models, and especially transformer-based language models, have shown to contain common sense and factual knowledge [58, 30]. Most recent multi-modal models have also relied on the transformer architecture to learn vision-and-language alignment [40, 46]. We adopt this direction in our work and build a multi-modal transformer model, pretrained with BERT [16], which has been pretrained on the following language corpora to capture implicit knowledge: BooksCorpus [86] (800M words) and English Wikipedia [1] (2.5B words). To learn multi-modal knowledge from the training set, our model is most closely related to the architecture used in [40]. We also explore multi-modal pretraining in Section 4.2.

Question Encoding. We tokenize a question Q using WordPiece [78] as in BERT [16], giving us a sequence of $|Q|$ tokens and embed them with the pretrained BERT embeddings and append BERT's positional encoding, giving us a sequence of d-dimensional token representation $x_1^Q, ..., x_{|Q|}^Q$. We feed these into the transformer, finetuning the representation during training.

Visual Features. As with most VQA systems, we use visual features extracted on the dataset by a visual recognition system trained on other tasks. We use bottom-up features [4] collected from the classification head of a detection model, specifically Faster R-CNN [61]. Because of the overlap in OK-VQA test and VisualGenome/COCO [42] trainval, we trained our detection model from scratch on VisualGenome, using a new split of VisualGenome not containing OK-VQA test images. The detector uses feature pyramid networks [43], and is trained using the hyperparameters used for the baselines in [29].

We input bounding box features extracted from the image as well as the question words to the transformer. We mean-pool the output of all transformer steps to get our combined implicit knowledge representation $z^{implicit}$.

3.2. Reasoning with Symbolic Knowledge

Visual Symbols. In addition to using a pretrained visual recognition system to get image features, we also extract visual concepts (i.e. the predictions). This not only allows us to get a set of concepts to use to prune our knowledge graph (see Sec. 3.2), it also gives us an entry point to get from the raw image to a set of symbols. This is significant—in order for our graph network to be able to reason about the question, it not only needs to reason about the question itself, but the entities in the image. For instance, if a question were to ask "what is a female one of these called?" in order use our knowledge that a female sheep is called an "ewe," the graph network needs to actually know that the thing in the picture is a sheep. As we will see, using these symbols is critical for our graph network to reason about the question.

There are a number of visual concepts we want to cover: places, objects, parts of objects and attributes. Therefore we run four classifiers and detectors trained on images from the following datasets: ImageNet [64] for objects, Places365 [83] for places, LVIS [28] for objects and object parts and Visual Genome [36] for objects, parts and attributes. This gives us a total of about 4000 visual concepts. (Additional details in supplementary).

Knowledge Graph Construction. Unlike previous work such as [54], or in NLP work on datasets such as SQuAD [59] which study the problem of closed-system knowledge retrieval, we do not have a ground truth set of facts or knowledge which can be used to answer the question. We must make an additional choice of what knowledge sources to use and how to clean or filter them.

There are a few different kinds of knowledge that might help us on this task. One is what one might call trivia knowledge: facts about famous people, places or events. Another is commonsense knowledge: what are houses made of, what is a wheel part of. Another is scientific knowledge: what genus are dogs, what are different kinds of nutrients. Finally, situational knowledge: where do cars tend to be located, what tends to be inside bowls.

The first and largest source of knowledge we use is DBPedia [7], containing millions of knowledge triplets in its raw form. DBPedia is created automatically from data from Wikipedia [1]. This tends to give a lot of categorical information e.g. (Denmark, is_a, country), especially about proper nouns such as places, people, companies, films etc. The second source of knowledge is ConceptNet [44], a crowd-sourced project containing over 100,000 facts organized as knowledge triples collected by translating English-language facts into an organized triplet structure. It also contains as a subset the WordNet [53] ontology. This dataset contains commonsense knowledge about the world such as (dog, has_property, friendly). Following [52], we

Relation Types

	has part	is a	used for	has a	at location	has property	located near	instance of	related to	made of	part of	capable of	causes	is on	is in	has	is made of	is at	is part of	is near	is for
hasPart KB	×																				
DBPedia	×	×																			
ConceptNet		×	×	×	×	×	×	×	×	×	×	×	×								
VisualGenome														×	×	×	×	×	×	×	×

Example Knowledge

hasPart KB	DBPedia	ConceptNet	VisualGenome
(bear, has part, coat)	(poland, is a, country)	(saloon, used for, drink)	(tree, is near, building)
(wasp, has part, wing)	(mark, is a, currency)	(stream, at location, forest)	(car, is on, road)
(cnidarian, has part, cell)	(easyjet, is a, company)	(eye, used for, look)	(building, is made of, bricks)
(alfalfa plant, has part, leave)	(gerbera, is a, insect)	(tearoom, used for, drink tea)	(outlet, is on, wall)
(water, has part, water molecule)	(new era, is a, automobile)	(heifer, at location, barnyard)	(tracks, is for, train)
(human, has part, bone)	(brussels, has part, ixelles)	(quartz, is a, mineral)	(chair, is near, table)
(hare, has part, long ear)	(syrah, is a, grape)	(star, at location, galaxy)	(food, is in, bowl)
(fern, has part, spore)	leona, is a, ship)	(hotel room, used for, sleep in)	(giraffe, has, spots)

Figure 2. Example knowledge and edge types from our knowledge graph. The graph is built from four sources of explicit knowledge.

also use the scene graphs from VisualGenome [36] as another source of knowledge. As in [52], we take a split of VisualGenome that does not contain any OK-VQA test images. This knowledge source tends to give us more spatial relationships e.g. (boat, is_on, water) and common pairwise affordances e.g. (person, sits_on, coach). Finally, we use the new hasPart KB [10] to get part relationships between common objects such as (dog, has_part, whiskers) as well as scientific ones (molecules, has_part, atoms). We show example knowledge triplets from our in Fig. 2.

With these knowledge sources, we can capture a large amount of knowledge about the world. But we then run into a problem of scale. In its raw form, DBPedia alone contains millions of edges, with the others containing a total of over 200,000 knowledge triplets. This first presents a technical problem—this graph is far too large to fit into GPU memory if we use a graph neural network model. But more fundamentally, while this knowledge graph contains a lot of useful information for our downstream task, it also includes a lot of irrelevant knowledge. In particular, DBPedia, being parsed automatically from Wikipedia pages, contains information about virtually every film, book, song and notable human in history. While some of those may be useful for particular questions, the vast majority is not.

To deal with these issues, we limit our knowledge graph to entities that are likely to be helpful for our end task. First, we collect all of the symbolic entities from the dataset: in particular the question, answers and visual concepts that can be picked up by visual recognition systems (see Sec. 3.2). We then include edges that only include these concepts. After this filtering, we have a total of about 36,000 edges and 8,000 nodes. We provide more exhaustive details of our knowledge collection and filtering in supplementary.

Graph Network. Now we move to our symbolic knowledge representation. We want to treat our knowledge graph as input without having to decide on which few facts out of our entire graph might be relevant. So to process on our entire graph and decide this during training, we use a graph neural network to incorporate our knowledge. In our network, each node of the graph network corresponds to one specific symbol representing one concept such as "dog" or "human" in our knowledge graph.

The idea is that the graph neural network can take in information about each specific symbol and use the knowledge edges to infer information about other symbols by passing information along the edges in the knowledge graph. And, in our graph neural network we share the network parameters across all symbols, meaning that unlike for other types of networks, the reasoning logic is shared across all symbols which should allow it to generalize better to rare symbols or graph edges.

We use the Relational Graph Convolutional Network (RGCN) [66] as the base graph network for our model. Unlike the related GCN [34], this model natively supports having different calculations between nodes for different edge types (an is_a relationship is treated differently than a has_a relationship) and edge directions (dog is_a animal is different than animal is_a dog). With this architecture we also avoid the large asymptotic runtime of other architectures with these properties such as [41] or [72].

Graph Inputs. For one particular question image pair, each node in the graph network receives 4 inputs. 1) An indicator $0/1$ of whether the concept appears in the question. 2) The classifier probabilities for the node's concept, introduced above (or 0 if the concept is not detected in the particular image or not one of the classifier's concepts) With 4 image classifiers or detectors, the node receives 4 separate numbers. 3) The $300d$ word2vec (GloVe [57]) representation of that concept, or average word2vec for multi-word concepts. 4) The implicit knowledge representation $z^{implicit}$ from Sec. 3.1 passed through a fully connected layer: $fc(z^{implicit})$ with ReLU activation to reduce the size of this feature to 128 for efficient graph computation.

Following the standard formulation of graph neural networks, we write the input to the graph neural networks (described above) as $X{=}H^{(0)}$ where X is a $\mathbb{R}^{n \times d_s}$ matrix with n node inputs of size $d_s = 433$. Then for each layer of the RGCN, we have a non-linear function $H^{(l+1)}{=}f(H^{(l)}, KG)$ where KG is the knowledge graph. The RGCN convolution uses different weight matrices for different edge types and for different directions. As a result the semantic difference between an is-a relationship and a has-a relationship as well as the direction of those edges is captured in the structure of the network and different transformations are learned for each. After all RGCN layers are computed we end up with $H^{(L)}{=}G$ which is a $\mathbb{R}^{n \times d_h}$ matrix which corresponds to having a hidden state of size f_h for each node (and therefore concept) in our graph. Additional architectural details and parameters of the graph network can be found in supplementary.

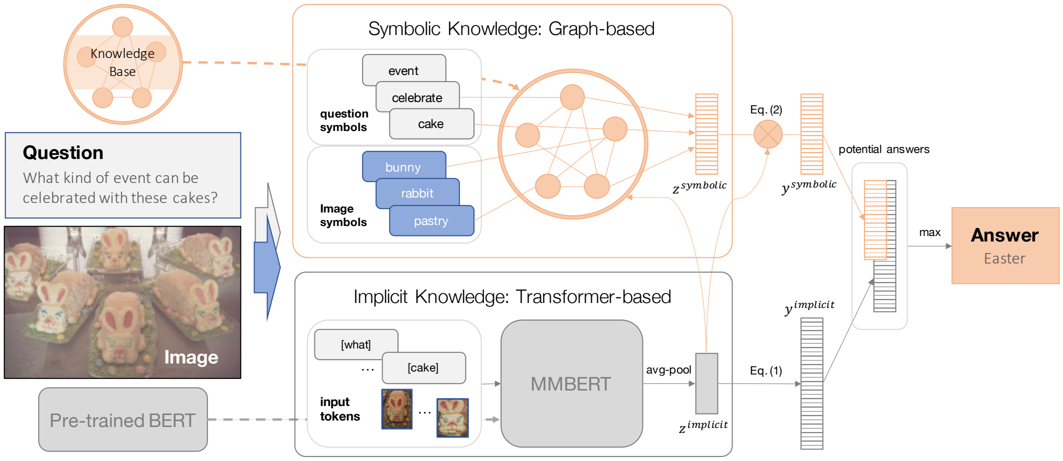


Figure 3. Our model: KRISP integrates implicit knowledge and reasoning (bottom) with explicit graph-based reasoning on a knowledge base (top). The implicit knowledge model receives the visual features and question encoding whereas the explicit knowledge model operates on image and question symbols. They predict answers according to Eq. 1&2 and we take the max overall prediction (see Sec. 3.3).

3.3. Integrating Implicit and Symbolic Knowledge

Finally, given the output of our implicit transformer-based module $z^{implicit}$ and our explicit/symbolic module G, how do we get our final prediction? Our main insight to make a separate prediction for $z^{implicit}$ and for each node/concept in the knowledge graph.

Implicit Answer Prediction. As is now commonplace among VQA methods, to get the implicit answer prediction, we do a final prediction layer and predict the answer within a set vocabulary of answers $V \in \mathbb{R}^a$ where a is the size of the answer vocabulary. We simply have:

$$y^{implicit} = \sigma(W z^{implicit} + b) \qquad (1)$$

where σ is the sigmoid activation.

Symbolic Answer Prediction. To predict the answers for symbolic, we note that G can be rewritten as a hidden state node $z_i^{symbolic}$ for each node/concept i in the knowledge graph. Because each of these nodes corresponds to a word or multi-word symbol, we actually have nodes and corresponding hidden states that are possible answers to a VQA question. So for each hidden state that is in our answer vocab $V \in \mathbb{R}^a$ we make a prediction for it.

For each of these answer nodes i, we predict:

$$y_i^{symbolic} = \sigma((W^s z_i^{symbolic} + b^s)^T (W^z z^{implicit} + b^z)). \qquad (2)$$

We additionally re-use the implicit hidden state $z^{implicit}$ to make this prediction. This gives us an additional late fusion between the implicit and symbolic parts of our model.

Final Prediction. Finally, given our final predictions $y^{implicit}$ and $y^{symbolic}$, we simply choose the final answer by choosing the highest scoring answer from both answer vectors. For training, we can simply optimize $y^{implicit}$ and $y^{symbolic}$ separately with a binary cross entropy loss end-to-end through the entire network. See Fig. 3.

4. Results

4.1. Experimental Setup

For all experiments, we train our models with PyTorch [56] and the MMF Multimodal Framework [68]. We use PyTorch Geometric [21] for our graph neural network implementations. We use the default training hyperparameters from MMF which we provide in supplementary. For consistency, for each result we train each model on 3 random seeds and take the average as the result. We show sample std on these runs in supplementary.

For the purpose of state-of-the art comparisons in Table 1, we compare our main method on the 1.0 version of OK-VQA [51]. Recently, a 1.1 version of the dataset was released, and all other experiments including ablations are done on this version. The only change between the versions is a change in how answer stemming is handled, resulting in a more coherent answer vocabulary. In particular, we observe that the new answer vocabulary has much fewer "non-word" stemming such as "buse" for busses and "poni tail" instead of "pony tail." Unless otherwise stated, an experiment is on version 1.1.

For many of our ablations and analysis we train just the Multi-modal BERT (MMBERT) model described in Sec. 3.1 by itself by scratch or we do multi-modal pretraining. Unless otherwise stated, this model and ours is always initialized from BERT.

In Sec. 4.3 we do a through ablation of KRISP comparing the different parts of the model and design choices we made. In Sec. 4.2 we add multimodal pretraining to our models to show how our model achieves state-of-the-art performance on OK-VQA. In Sec. 4.4 we show the results of a number of experiments to more thoroughly analyze our method, especially looking at its performance on rare answers. Finally in Sec. 4.5 we look at some specific questions and predictions from our model to get a more grounded idea

Method	accuracy (v1.0)	accuracy (v1.1)
Q-Only	14.93	-
MLP	20.67	-
BAN [33]	25.17	-
BAN+AN [51]	25.61	-
BAN+KG-Aug [39]	26.71	-
MUTAN [8]	26.41	-
MUTAN+AN [51]	27.84	26.64
ConceptBERT [26]	33.66	-
KRISP (w/o mm pre.)	29.77	32.31
KRISP (with mm pre.)	**38.35**	**38.90**

Table 1. Benchmark results on OK-VQA

of what our model does on real examples.

4.2. State-of-the-Art Comparisons

We provide the comparisons to the state-of-the-art of OKVQA in Table 1. To achieve best results, like other works [26] we pretrain our network on other tasks. We find it the most effective to pretrain our models on the VQA dataset [27]. See supplementary for more details.

In order to compare to other works (all of which show results on v1.0), we compute the performance of our best model (VQA joint graph and transformer pretraining) on OK-VQA v1.0 as well. We see that our model achieves 38.35% accuracy versus the best previous state-state-of-the-art of 33.66% [26]. We also compare on v1.1 as well, rerunning the MUTAN+AN model from [51] to get a comparison with KRISP.

4.3. Model Analysis and Ablations

We first analyse our model to see where the improvement is coming from with several ablations, especially focusing on symbolic *vs.* implicit knowledge and their integration. We want to understand which parts are working and why.

Ablation of Symbolic Knowledge. First, we see how much of the improvement comes from the Multi-modal BERT backbone of our model versus from the symbolic Graph Network. In Table 2 (lines 1&2), we see that KRISP combining implicit and symbolic knowledge improves significantly over the Multi-modal BERT by about 3%.

We should, however, make sure this improvement is due to the symbolic knowledge and not merely from a more complex or better architecture. While our KRISP only has slightly more parameters (116M parameters versus MMBERT with 113M), it does add at least some extra computation. To test this, we approximate a version of our method with only the architecture and not the underlying knowledge. To do this, we keep all network details the same, but instead of using the knowledge graph we constructed in Sec. 3.2, we use a randomly connected graph. We keep all of the nodes the same, but we randomize the edges connecting them. So in this version with a random graph, our graph network receives all of the same inputs and the outputs, but all connections are completely random. If the performance were just from the computation, we would expect this to work. Instead, we see from line 3 that the performance using the random graph drops significantly.

	Method	accuracy
1.	KRISP (ours)	**32.31**
	Ablation of Symbolic Knowledge	
2.	MMBERT	29.26
3.	KRISP w/ random graph	30.15
	Ablation of Implicit Knowledge	
4.	KRISP w/o BERT pretrain	26.28
5.	MMBERT w/o BERT pretrain	21.82
	Ablation of Network Architecture	
6.	KRISP no late fusion	31.10
7.	KRISP no MMBERT input	31.10
8.	KRISP no MMBERT input or late fusion	25.00
9.	KRISP no backprop into MMBERT	27.98
10.	KRISP with GCN	30.58
11.	KRISP feed graph into MMBERT	30.99
	Ablation of Graph Inputs	
12.	KRISP no Q to graph	31.74
13.	KRISP no I to graph	31.59
14.	KRISP no symbol input	30.26
15.	KRISP no w2v	31.95

Table 2. KRISP ablation on OK-VQA v1.1. We show the performance of our model compared with the implicit-only baseline (MMBERT). We also show ablations without BERT training, with a random knowledge graph, ablations on our model architecture, and ablations where we remove the question input to the graph network (no Q), the image inputs (no I) and both (no symbol).

Ablation of Implicit Knowledge. Next we look at the implicit knowledge contained in the BERT versus our combined system to see how much of an effect it had. From Table 2 we can see that BERT is a crucial element. Without the BERT pretraining (lines 4&5), our method falls by 6% and the Multi-modal BERT falls by an even larger 7%. This shows that the implicit knowledge is an important component of our model. The difference between KRISP and Multi-modal BERT when neither has BERT pretraining is actually higher than the difference with BERT, about 4.5%, suggesting that there is some overlap in the knowledge contained in our knowledge graphs with the implicit knowledge in BERT, but most of that knowledge is non-overlapping.

Ablation of Network Architecture. Next, we want to get a sense of which parts of our architecture were important. As we can see, our particular architecture is critical: the use of MMBERT features as input to KRISP and the late fusion were both important. With just one of these, performance drops by about 1%, but without either (line 8), performance drops over 7%. Without at least one connection between the Multi-modal BERT and the graph network, there can be no

	Method	accuracy
1.	KRISP $\max(y^{implicit}, y^{symbolic})$ (ours)	**32.31**
2.	KRISP $y^{implicit}$	31.47
3.	KRISP $y^{symbolic}$	29.36
4.	KRISP no backprop $y^{implicit}$	28.19
5.	KRISP oracle$(y^{implicit} \vert y^{symbolic})$	36.71

Table 3. KRISP Subpart Analysis on OK-VQA v1.1. Here we show the OK-VQA accuracy of different parts of the model separately: just the MMBERT ($y^{implicit}$), just the graph network ($y^{symbolic}$). We also show the MMBERT only without a backpropogation signal between the two parts and an oracle best-case performance between the two parts.

fusion of the visual features and question and the graph network cannot incorporate any of the implicit knowledge in BERT. We also tried KRISP where these two ways of fusing were present, but we did not allow any backpropogation from the Graph Network to MMBERT (line 9). This also performs badly, as the graph network cannot correct errors coming from this input, but not as bad as removing these connections entirely (line 8).

We also tried a less powerful graph network: GCN [34] (line 10) which critically does not have directed edges or edge types. This baseline hurts performance by about 2% justifying our choice of a graph network that uses edge direction and type. We also have another architectural ablation, where we feed the graph network features directly to the Multi-modal BERT rather than having a separate answer prediction directly from the graph as in KRISP or any of the other baselines (line 11). This architecture performs much worse than our final model.

Ablation of Graph Inputs. Next we look at the symbolic and non-symbolic inputs to the knowledge graph nodes to see what effect those might have had in the next section of Table 2. First, we ablate the question indicator input (line 12) and the image confidences (line 13) described in Sec. 3.2. We find that removing one or the other drops performance, but not drastically; removing both (line 14) drops performance by about 2%, much more than the effect of dropping the MMBERT input to the graph. We also ablate the word2vec inputs to nodes (line 15) and find that this part made the least difference, dropping it less than 1%.

Preserving Symbolic Meaning. One major claim we make is that symbolic and implicit knowledge are both necessary for this problem. The results without BERT training make the case pretty clearly that implicit, non-symbolic knowledge from BERT is critical. From the ablation of symbolic knowledge, we show that it is the symbolic knowledge (and not just the architecture) greatly contributes to the performance of our method. On the symbol input side, we show that removing the symbolic inputs (line 12) hurts performance, even more than removing the Multi-modal BERT hidden input (line 7) which contains information about the same image and question, but in a non-symbolic form. Finally we have a baseline (line 11) where instead of predicting separate outputs from the graph network and Multi-modal BERT, we directly connect the graph network into MMBERT, feeding a pooled graph hidden state (see supplementary for details) into MMBERT as an input. This baseline does significantly worse. What these ablations have in common is that they remove the direct connection between the knowledge graph and the input and/or answer symbols. When the graph network is not able to connect the knowledge symbolically to the input symbols or the output symbols, we see that it performs worse. In addition, we know symbolic knowledge itself is useful because when we only change the connections between nodes and nothing else (line 3), performance drops drastically. Our entire graph module directly connects symbols in the input (question words and image symbols from classifiers) to symbols in the output (the answer words) and this seems critical to performance.

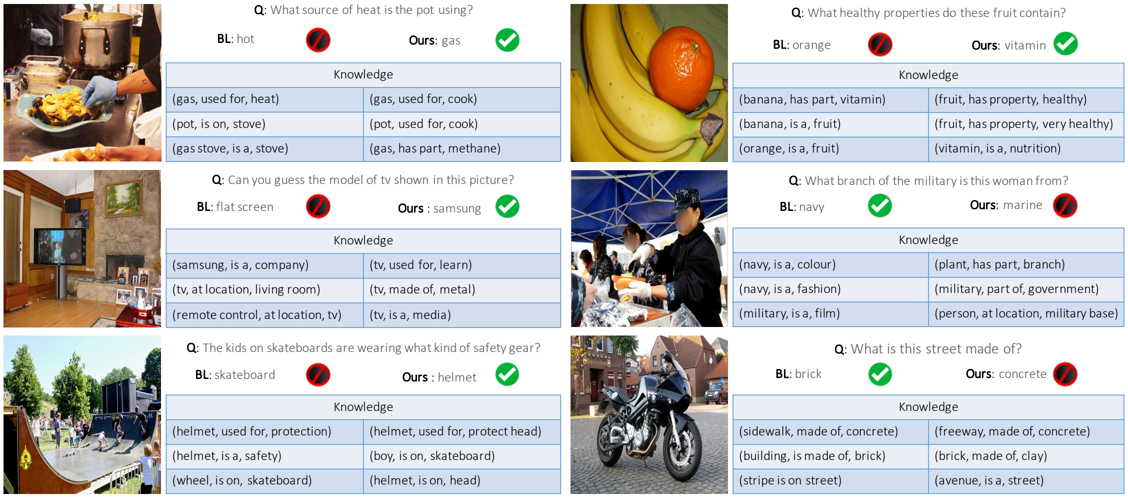
Metric→	Frequency Rank		# Unique answers	
Method ↓	All	Correct	All	Correct
KRISP (ours)	**528.5**	**456.7**	**1349**	**780**
MMBERT	467.1	427.4	1247	719

Table 4. Long-tail Analysis. We show KRISP and the non-symbolic MMBERT long-tail metrics for "all" predictions made by the model and for "correct" predictions. Higher is better.

4.4. Quantitative Result Analysis

First we examine the parts of our model separately to see if we can learn anything about how the MMBERT and Graph Network parts of KRISP interact.

In Table 3 we look at the performance of different parts of our model (without retraining the model for lines 1,2,3,5). Since the MMBERT and Graph Network parts of KRISP produce separate predictions, we can analyze them separately. For instance, we find that despite the fact that the MMBERT part of our model does not receive input from the Graph Network, the MMBERT (Table 3, line 2) has a higher accuracy of 31.47% than the MMBERT baseline (Table 2, line 2), 29.26%. This we suspect is because this part of the network receives a back-propagation from the Graph Network part of the model and this extra component improves the quality of the MMBERT pooled feature because it is also trained to reduce the loss from the late fusion predictions. Indeed, if we remove the back-propagation signal (Table 3, line 4) we see that the accuracy of this part of the model drops down to 28.19%. We also see a direct improvement beyond this effect. Comparing the Multi-modal BERT (line 2) and Graph Network (line 3) -only accuracies, the Graph Network does a bit worse on its own, but not by a huge amount, and the Graph Network predictions are used 47% of the time in the joint model (line 1). Since the accu-



Q: What source of heat is the pot using?

BL: hot ⊘ Ours: gas ✓

Knowledge	
(gas, used for, heat)	(gas, used for, cook)
(pot, is on, stove)	(pot, used for, cook)
(gas stove, is a, stove)	(gas, has part, methane)

Q: What healthy properties do these fruit contain?

BL: orange ⊘ Ours: vitamin ✓

Knowledge	
(banana, has part, vitamin)	(fruit, has property, healthy)
(banana, is a, fruit)	(fruit, has property, very healthy)
(orange, is a, fruit)	(vitamin, is a, nutrition)

Q: Can you guess the model of tv shown in this picture?

BL: flat screen ⊘ Ours : samsung ✓

Knowledge	
(samsung, is a, company)	(tv, used for, learn)
(tv, at location, living room)	(tv, made of, metal)
(remote control, at location, tv)	(tv, is a, media)

Q: What branch of the military is this woman from?

BL: navy ✓ Ours: marine ⊘

Knowledge	
(navy, is a, colour)	(plant, has part, branch)
(navy, is a, fashion)	(military, part of, government)
(military, is a, film)	(person, at location, military base)

Q: The kids on skateboards are wearing what kind of safety gear?

BL: skateboard ⊘ Ours : helmet ✓

Knowledge	
(helmet, used for, protection)	(helmet, used for, protect head)
(helmet, is a, safety)	(boy, is on, skateboard)
(wheel, is on, skateboard)	(helmet, is on, head)

Q: What is this street made of?

BL: brick ✓ Ours: concrete ⊘

Knowledge	
(sidewalk, made of, concrete)	(freeway, made of, concrete)
(building, is made of, brick)	(brick, made of, clay)
(stripe is on street)	(avenue, is a, street)

Figure 4. Qualitative examples from KRISP. Showing predictions by our model and the implicit knowledge baseline Multi-modal BERT. We show the question, image, and answers given by both models. We also show knowledge in the graph related to the question, answers or image that seemed most relevant.

racy of the combined model is higher than each, it is able to choose the correct answer from between MMBERT and Graph Network. Finally, we see that if we had an oracle that always chose the best prediction from either the MMBERT or the Graph Network, we would improve the accuracy to 36.71%. Obviously this is not a realistic number to achieve since it uses ground truth, but it shows that the MMBERT and Graph Network predictions are non-redundant.

Long-Tail Analysis. Next, we try to see whether our explicit/implicit model performs any differently on the "long tail" of OK-VQA. OK-VQA itself is built as a long-tail dataset, specifically rejecting answers that appear too many times to avoid models overfitting to the answer vocabulary, making it a good dataset to study knowledge-based VQA. Even with this filtering, some answers do appear more often than others, so we can try to study whether our method does better on rare answers.

In Table 4 we show metrics on KRISP versus the baseline Multi-modal BERT. First we use a metric we refer to as "Answer Frequency Rank". This simply means we order the answers in the dataset from most common to least common and assign them a rank from 1 for the most common to the total number of answers in the dataset. On this metric our model scores higher, which means it chooses on average less common answers. This is true whether one measures for all prediction or for only correct predictions. For a perhaps more intuitive metric we also look at the number of unique answers our model predicts versus the baseline. Here we predict 1349 versus 1247 or 780 versus 719 if we only look at correct predictions. These results indicate that our model is generalizing better to the long-tail.

4.5. Qualitative Analysis

Finally, we show examples to understand how the knowledge graph might be helping our model to answer questions. In the top left example in Fig. 4 our model correctly answers that the source of heat for the pot is "gas." Looking at the knowledge graph, some knowledge that may be helpful is that gas is used for heat, and that both gas and pot are used to cook. The knowledge graph here connects directly from a word in the question to the answer. The next question asks what model the TV is and our model predicts Samsung. This is supported by an edge that indicates that Samsung is a company which makes it more likely to be a "model" of a product. We include more examples in supplementary.

5. Conclusion

In this paper we introduce *Knowledge Reasoning with Implicit and Symbolic rePresentations* (KRISP): a method for incorporating implicit and symbolic knowledge into Knowledge-Based VQA. We show it outperforms prior works on OK-VQA [51], the largest available open-domain knowledge VQA dataset. We show through extensive ablations that our particular architecture outperforms baselines and other alternatives by preserving the symbolic representations from input to prediction. Moreover, through experiments, analysis, and examples we find our model makes use of both implicit and symbolic knowledge to answer knowledge-based questions and generalizes to rare answers.

Acknowledgements: We want to thank Amanpreet Singh, Abhishek Das, Ronghang Hu and Vedanuj Goswami who provided data and help with the MMF Multimodal Framework and PyTorch Geometric. As part of their affiliation with CMU, Kenneth Marino is supported by DoD through the NDSEG Program. The Georgia Tech effort was supported in part by NSF, AFRL, DARPA, and ONR YIP. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the U.S. Government, or any sponsor.

References

[1] Wikipedia: The free encyclopedia. https://www.wikipedia.org/.

[2] Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C Lawrence Zitnick, Devi Parikh, and Dhruv Batra. Vqa: Visual question answering. *IJCV*, 2017.

[3] Chris Alberti, Jeffrey Ling, Michael Collins, and David Reitter. Fusion of detected objects in text for visual question answering. In *EMNLP*, 2019.

[4] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, pages 6077–6086, 2018.

[5] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks. In *CVPR*, 2016.

[6] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: visual question answering. In *ICCV*, 2015.

[7] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer, 2007.

[8] Hedi Ben-younes, Rémi Cadene, Matthieu Cord, and Nicolas Thome. Mutan: Multimodal tucker fusion for visual question answering. In *ICCV*, 2017.

[9] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic parsing on freebase from question-answer pairs. In *EMNLP*, 2013.

[10] Sumithra Bhakthavatsalam, Kyle Richardson, Niket Tandon, and Peter Clark. Do dogs have whiskers? a new knowledge base of haspart relations. *arXiv preprint arXiv:2006.07510*, 2020.

[11] Antoine Bordes, Sumit Chopra, and Jason Weston. Question answering with subgraph embeddings. In *EMNLP*, 2014.

[12] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading wikipedia to answer open-domain questions. In *ACL*, 2017.

[13] Xinlei Chen, Li-Jia Li, Li Fei-Fei, and Abhinav Gupta. Iterative visual reasoning beyond convolutions. In *CVPR*, pages 7239–7248, 2018.

[14] Xinlei Chen, Abhinav Shrivastava, and Abhinav Gupta. Neil: Extracting visual knowledge from web data. In *ICCV*, 2013.

[15] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Learning universal image-text representations. *arXiv preprint arXiv:1909.11740*, 2019.

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[17] Santosh Divvala, Ali Farhadi, and Carlos Guestrin. Learning everything about anything: Webly-supervised visual concept learning. In *CVPR*, 2014.

[18] Kun Duan, Devi Parikh, David Crandall, and Kristen Grauman. Discovering localized attributes for fine-grained recognition. *CVPR*, 2012.

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

[20] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *TPAMI*, 28, 2006.

[21] Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. *arXiv preprint arXiv:1903.02428*, 2019.

[22] Andrea Frome, Greg S. Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. *NeurIPS*, 2013.

[23] Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *EMNLP*, 2016.

[24] Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *EMNLP*, pages 457–468, 2016.

[25] Peng Gao, Zhengkai Jiang, Haoxuan You, Pan Lu, Steven CH Hoi, Xiaogang Wang, and Hongsheng Li. Dynamic fusion with intra-and inter-modality attention flow for visual question answering. In *CVPR*, pages 6639–6648, 2019.

[26] François Gardères, Maryam Ziaeefard, Baptiste Abeloos, and Freddy Lecue. Conceptbert: Concept-aware representation for visual question answering. In *EMNLP*, pages 489–498, 2020.

[27] 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 *CVPR*, 2017.

[28] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019.

[29] Huaizu Jiang, Ishan Misra, Marcus Rohrbach, Erik Learned-Miller, and Xinlei Chen. In defense of grid features for visual question answering. In *CVPR*, 2020.

[30] Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. How can we know what language models know? *TACL*, 8:423–438, 2020.

[31] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David A Shamma, Michael S Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. *CVPR*, 2015.

[32] Daniel Khashabi, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. Unifiedqa: Crossing format boundaries with a single qa system. *arXiv preprint arXiv:2005.00700*, 2020.

[33] Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. Bilinear attention networks. In *NeurIPS*, pages 1564–1574, 2018.

[34] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[35] Satwik Kottur, Ramakrishna Vedantam, José MF Moura, and Devi Parikh. Visual word2vec (vis-w2v): Learning visually grounded word embeddings using abstract scenes. In *CVPR*, pages 4985–4994, 2016.

[36] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 123(1):32–73, 2017.

[37] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *TPAMI*, 2014.

[38] Gen Li, Nan Duan, Yuejian Fang, Daxin Jiang, and Ming Zhou. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. *arXiv preprint arXiv:1908.06066*, 2019.

[39] Guohao Li, Xin Wang, and Wenwu Zhu. Boosting visual question answering with context-aware knowledge aggregation. In *ACMMM*, New York, NY, USA, 2020. Association for Computing Machinery.

[40] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019.

[41] Yujia Li and Richard Zemel. Gated graph sequence neural networks. *ICLR*, 2016.

[42] T. Lin, M. Maire, S. J. Belongie, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: common objects in context. *ECCV*, 2014.

[43] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 2117–2125, 2017.

[44] Hugo Liu and Push Singh. Conceptnet—a practical commonsense reasoning tool-kit. *BT technology journal*, 22(4):211–226, 2004.

[45] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In *ECCV*, pages 852–869. Springer, 2016.

[46] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*, 2019.

[47] Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 12-in-1: Multi-task vision and language representation learning. In *CVPR*, pages 10437–10446, 2020.

[48] Shangwen Lv, Daya Guo, Jingjing Xu, Duyu Tang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, and Songlin Hu. Graph-based reasoning over heterogeneous external knowledge for commonsense question answering. In *AAAI*, volume 34, pages 8449–8456, 2020.

[49] Mateusz Malinowski and Mario Fritz. Towards a visual turing challenge. In *arXiv*, 2014.

[50] Mateusz Malinowski, Marcus Rohrbach, and Mario Fritz. Ask your neurons: A neural-based approach to answering questions about images. In *ICCV*, 2015.

[51] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *CVPR*, 2019.

[52] Kenneth Marino, Ruslan Salakhutdinov, and Abhinav Gupta. The more you know: Using knowledge graphs for image classification. In *CVPR*, 2017.

[53] George A. Miller. Wordnet: A lexical database for english. *ACM*, 38, 1995.

[54] Medhini Narasimhan, Svetlana Lazebnik, and Alexander G Schwing. Out of the box: Reasoning with graph convolution nets for factual visual question answering. *NeurIPS*, 2018.

[55] Medhini Narasimhan and Alexander G. Schwing. Straight to the facts: Learning knowledge base retrieval for factual visual question answering. In *ECCV*, 2018.

[56] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pages 8026–8037, 2019.

[57] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014.

[58] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. Language models as knowledge bases? In *EMNLP*, pages 2463–2473, Hong Kong, China, Nov. 2019. Association for Computational Linguistics.

[59] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *EMNLP*, 2016.

[60] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *CVPR*, pages 7263–7271, 2017.

[61] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *NeurIPS*, 2015.

[62] Marcus Rohrbach, Sandra Ebert, and Bernt Schiele. Transfer Learning in a Transductive Setting. In *NeurIPS*, 2013.

[63] Marcus Rohrbach, Michael Stark, and Bernt Schiele. Evaluating Knowledge Transfer and Zero-Shot Learning in a Large-Scale Setting. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2011.

[64] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3):211–252, 2015.

[65] Fereshteh Sadeghi, Santosh K Divvala, and Ali Farhadi. Viske: Visual knowledge extraction and question answering by visual verification of relation phrases. In *CVPR*, 2015.

[66] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pages 593–607. Springer, 2018.

[67] Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Constrained semi-supervised learning using attributes and comparative attributes. *ECCV*, 2012.

[68] Amanpreet Singh, Vedanuj Goswami, Vivek Natarajan, Yu Jiang, Xinlei Chen, Meet Shah, Marcus Rohrbach, Dhruv Batra, and Devi Parikh. Mmf: A multimodal framework for vision and language research. https://github.com/facebookresearch/mmf, 2020.

[69] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. In *ICLR*, 2019.

[70] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *EMNLP*, 2019.

[71] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017.

[72] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[73] Peng Wang, Qi Wu, Chunhua Shen, Anthony R. Dick, and Anton van den Hengel. Explicit knowledge-based reasoning for visual question answering. In *IJCAI*, 2017.

[74] Peng Wang, Qi Wu, Chunhua Shen, Anton van den Hengel, and Anthony R. Dick. Fvqa: Fact-based visual question answering. *TPAMI*, 2017.

[75] Shuohang Wang, Mo Yu, Xiaoxiao Guo, Zhiguo Wang, Tim Klinger, Wei Zhang, Shiyu Chang, Gerald Tesauro, Bowen Zhou, and Jing Jiang. R3: Reinforced reader-ranker for open-domain question answering. *arXiv preprint arXiv:1709.00023*, 2017.

[76] Xiaolong Wang, Yufei Ye, and Abhinav Gupta. Zero-shot recognition via semantic embeddings and knowledge graphs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[77] Qi Wu, Peng Wang, Chunhua Shen, Anthony R. Dick, and Anton van den Hengel. Ask me anything: Free-form visual question answering based on knowledge from external sources. In *CVPR*, 2016.

[78] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

[79] Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. End-to-end open-domain question answering with bertserini. *NAACL*, page 72, 2019.

[80] Yi Yang, Wen-tau Yih, and Christopher Meek. Wikiqa: A challenge dataset for open-domain question answering. In *EMNLP*, pages 2013–2018, 2015.

[81] Xuchen Yao and Benjamin Van Durme. Information extraction over structured data: Question answering with freebase. In *ACL*, 2014.

[82] Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Josh Tenenbaum. Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. In *NeurIPS*, pages 1031–1042, 2018.

[83] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *TPAMI*, 2017.

[84] Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason J Corso, and Jianfeng Gao. Unified vision-language pre-training for image captioning and vqa. *arXiv preprint arXiv:1909.11059*, 2019.

[85] Yuke Zhu, Alireza Fathi, and Li Fei-Fei. Reasoning about object affordances in a knowledge base representation. In *ECCV*, 2014.

[86] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *ICCV*, pages 19–27, 2015.

[87] Yuke Zhu, Joseph J. Lim, and Li Fei-Fei. Knowledge acquisition for visual question answering via iterative querying. In *CVPR*, 2017.

[88] Yuke Zhu, Ce Zhang, Christopher Ré, and Li Fei-Fei. Building a large-scale multimodal knowledge base for visual question answering. *arXiv*, 2015.