Knowledge Acquisition for Visual Question Answering via Iterative Querying

learn new skills and new knowledge for problem solving.
an automatic model to deal with arbitrary, open-ended questions in the visual world.
a neural-based approach to acquiring **task driven information** for visual question answering (VQA).
queries to actively acquire relevant information from external auxiliary data.
Supporting evidence from either human-curated or automatic sources is encoded and stored into a memory bank.
acquiring task-driven evidence effectively improves model performance on both the Visual7W and VQA datasets
these queries offer certain level of interpretability in our iterative QA model.

1. Introduction

daily interactions involve asking follow-up questions and collecting “clues” – in order to:
(1) communicate with others
(2) answer their questions
(3) serve their needs.
collect “clues” to solve VQA task?

VQA methods make predictions based on a predefined set of information:
mixed representation of the image and the question sentence [3, 9, 23, 39, 41].
shown to be “myopic” (fail on novel instances) [1].
state-of-the-art VQA models could benefit from better visually grounded evidence [9, 17].
enabling a VQA model to
(1) ask for
(2) collect “clues”
visually grounded evidence from:
(1) human-curated

(2) algorithmically generated
data sources.

deep learning-based models dominated standard VQA benchmarks [3, 25, 31, 41].
the most popular choices is to use CNN to encode images and LSTM to encode words [3, 26, 31].
attention mechanism adopted by top models [9, 23, 41] to achieve better results.

[17]
proposed a two-layer MLP takes answers as input and makes binary predictions.
This simple network has shown highly competitive results in comparison to other more complex architectures.

extend their model with iterative querying framework to
（1） gather
（2） reason about
supporting evidence to tackle the VQA tasks.
not all evidence is equally valuable.
most pieces of evidence are irrelevant, and only a few of them are helpful.
a model has to be selective – ask for and use only relevant information to the task.
a dynamic VQA model that can iteratively ask queries for new evidence and collect relevant evidence from external sources.
obtains supporting evidence through a series of queries from external knowledge sources.
The acquired evidence is encoded and added to a memory bank.
the model with updated memory propose another round of queries, or answer the target question.
the model can work well with both
(1) human-curated knowledge sources, such as Visual Genome scene graphs [20],
(2) algorithmically generated knowledge sources by the state-of-the-art object detectors [32].

the model achieves new state-of-the-art performance on the Visual7W telling task [41],
on par with the top-performing model [9] on the VQA Real Multiple Choice challenge.
Another advantage is its interpretability.
At every iteration, the model actively seeks new evidence with a textual query.
It enables us to examine the model's "rationale" in its iterative process of seeking the final answer.

2. Related Work

VQA Models.
(1) symbolic approaches [25, 38],
(2) neural-based approaches [9, 23, 24, 26, 31],
(3) hybrid scheme [2].
Attention mechanisms [9, 23, 39, 41]:
shown effective in fusing the multi-modal representations of the question words and the images.
the behavior of existing VQA models [1],
evaluating model attention maps against human attention [8].
[17] takes answers as input and performs binary predictions. competes with other complex VQA systems.
extends model [17] with a memory bank, better on Visual7W dataset [41] and VQA challenge [3].

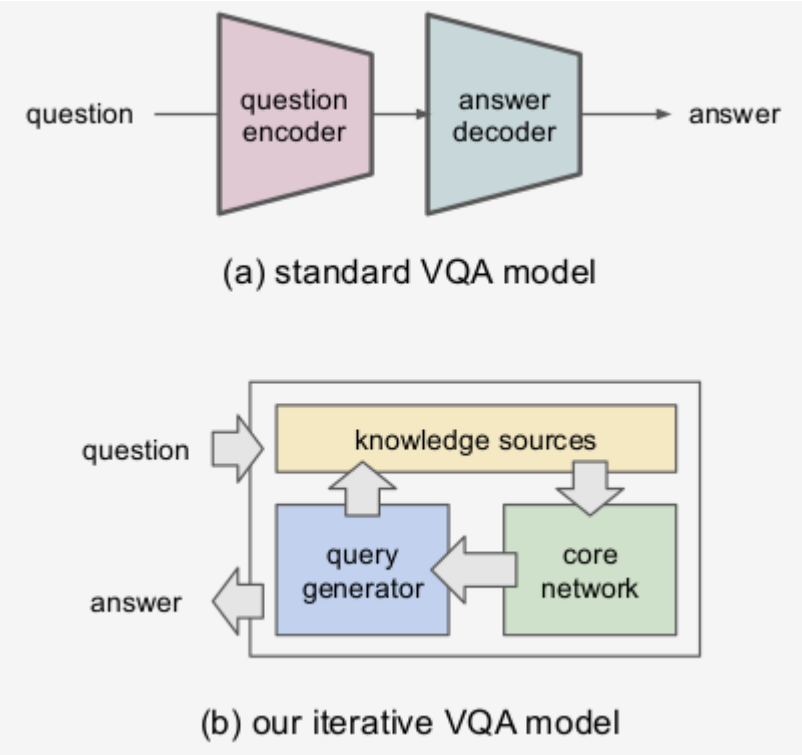
Interactive Knowledge Acquisition.
SHRDLU [36], provided a dialog system for users to query a computer about the state of a simplified blocks world.
developed interactive interfaces to

(1) acquire knowledge from human experts [37]
(2) efficiently label new training samples [35]
(3) propose the next question in a restricted visual Turing test [10]
never-ending learning:
(1) NELL [6]
(2) NEIL [7].
knowledge harvested in a never-ending loop is often arbitrary and suffers from semantic drift.
a variety of strategies to acquire knowledge from external sources [4, 14, 19, 29, 30, 34].

our work builds a neural-based framework, capable of handling multi-modal data.
Learn a query strategy in a data-driven fashion.

Memory Networks.
augmenting neural networks with memory.
LSTM [15] introduces memory cells to vanilla RNNs.
Recent work [11, 12, 18, 21, 33, 40] focuses on developing different external memory representations based on attention mechanism.

dynamic memory network [5]:
(1) has an episodic memory module to encode task-dependent information for VQA.
(2) encodes image features into the episodic memory module, not semantic information.
(3) does not learn a strategy to select task-driven evidence based on its relevance to the task.



3. Methods

Many visual questions require open-ended common sense reasoning [3, 10, 16, 41].
most models are “myopic” fail on sufficiently novel concepts.
Instead of learning within a closed set, requires a more flexible and principled model that learns and reasons with new information.

design a model proposes **queries** and acquires task-driven **evidence** from **knowledge** source
[[task ---> queries ---> knowledge source ---> evidence]]

focus on visually grounded evidence from:
(1) human-curated
(2) algorithmically generated
data sources.

introduce a model dynamically and constantly learns from external environments in a multi-step fashion.
The key challenge here is to:
learn a querying strategy to gather the most informative evidence for the task.

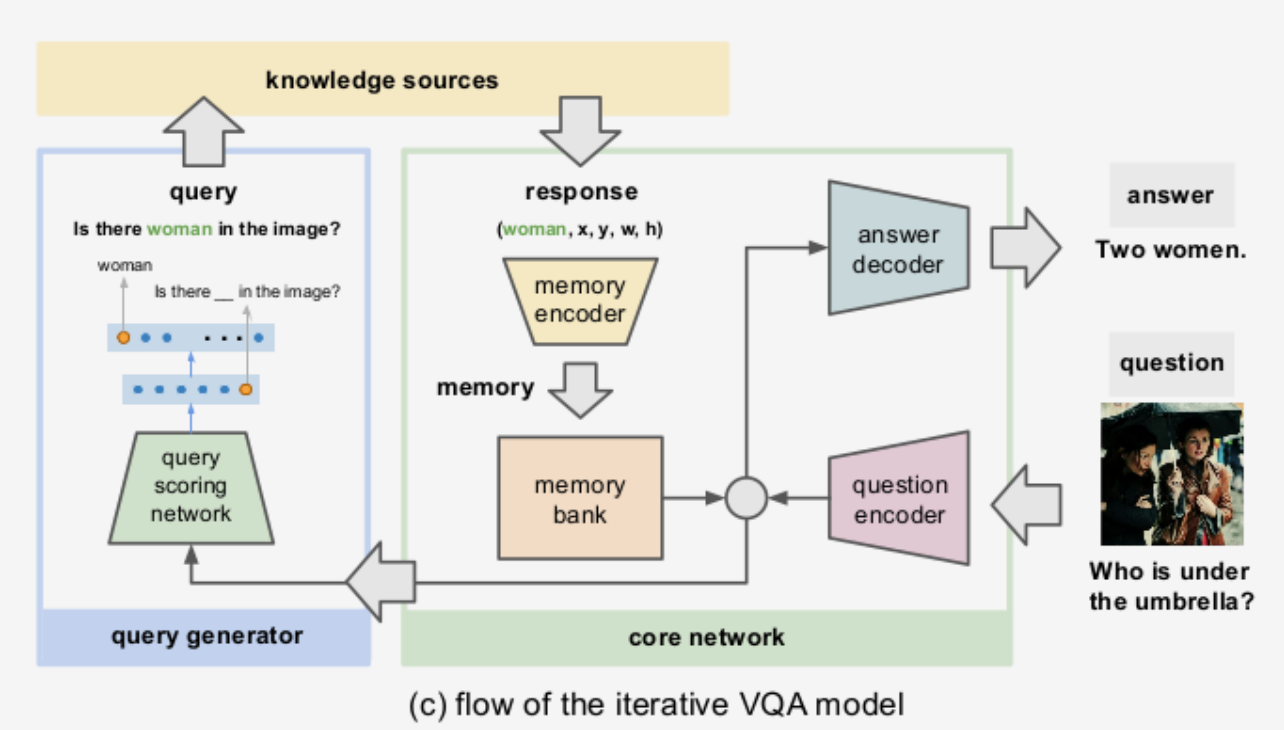
3.1. Model Overview

Goal： iteratively obtain **task-driven evidence** to produce an answer to a given visual question.
This process requires a model that can ask for necessary information from the external sources.
use query and response as the means of communication between the model and the knowledge sources.
the model encode new evidence from a response in its internal representation called memory.

Five data types	
question	the model aims to answer about the image. (***q, i***)： a question ***q*** on image ***i***
answer	a natural language answer to the question. the multiple choice tasks: select the correct one from a set of candidate answers
query	a sentence requesting for a piece of task-driven information by the model
response	a piece of evidence from knowledge sources to a query
memory	an encoded evidence. Raw evidence is encoded into a vector of memory that the model can store and process.

Two major challenges in designing an iterative VQA model:
1) proposing the next query at the current model state
2) updating the model state with acquired evidence, potentially in various forms from different sources.

Model	
core network	updating the memory state and generating an answer
query generator	proposing the next query based on the memory state



(c) flow of the iterative VQA model

3.2. Core Network

(1) takes a question input,
(2) predicts an answer
(3) maintains its internal memory bank
(4) iteratively obtaining new evidence through querying.

4 sub-networks			
memory encoders	**memory bank**	**question encoder**	**answer decoder**
f_K	M	E_q	G_a
raw evidence e to a memory vector that the memory bank can store and process: $m=f_K(e)$ Raw evidence can be: (1) heterogeneous (2) multi-modal different types of evidence into vectors of the same size;	stores a collection of memories acquired via iterative querying, where $M=\{m^{(1)},m^{(2)},...,m^{(t)}\}$. supports both read/write operations: (1) generate a representation of current memory state $\varphi_M(read)$. (2) a new memory encoded and added to memory bank, where $M:=M \cup \{e^{(t+1)}\}$ $(write)$;	encodes a question-image pair into a vector embedding $v=E_q(q, i)$;	takes the question encoding v and the memory state φ_M, and produces an answer $a=G_a(v, \varphi_M)$. The question encoder and answer decoder can also be coupled in a single network [17]. each sub-network is modularized. the effectiveness of our model even without complex network design.
transforms raw evidence from external knowledge sources into fixed-dimensional memory vectors.	stack	two-layer MLP model [17]	two-layer MLP model [17]

represent each memory as a 300-dimensional averaged word2vec embedding [27].	keeps the encoded memory vectors updates itself by adding a new memory to the stack.	input a concatenation of: (1) pre-trained image features [13], (2) an average of word embeddings of the question and the answers,	
	compute the memory state by summing the memory vectors, normalized by l_2-norm, in the memory bank.	predicts whether an image-question-answer triplet is correct.	
	concatenate this memory state vector with the image-question-answer triplet as input to MLP model.		

3.3. Query Generator

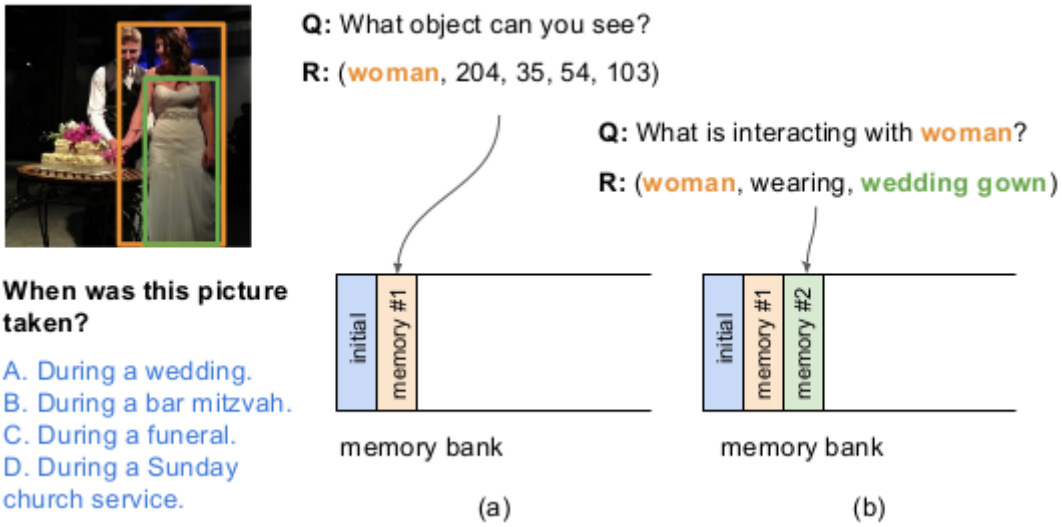


Figure 3: An example iterative query process. At each time, the model proposes a task-driven query (Q) to request useful evidence from knowledge sources. The response (R) is encoded into a memory and added to the memory bank.

The query generator：

bridges model with the external knowledge sources.

proposes queries based on the memory state to obtain best relevant evidence.

the most straightforward strategy：

(1) paraphrase the target question as a query to an omniscient source

(2) no such oracle in practice.

it is essential to:

(1) define useful query types for communication

(2) devise a good query strategy for effectiveness.

visual grounding: facts about the objects in an image
[17] indicates that:
  a lack of visual grounding is a key problem in current VQA systems
visual grounding help resolving the underlying uncertainty from noisy vision models.

4 query types the model can use to request visually grounded evidence.

Table 1: Query Types and Response Formats

Query types and templates	**Response formats**
What object can you see?	(**object**, x, y, w, h)
Is there **object** in the image?	(**object**, x, y, w, h)
How does **object** look?	(**object**, **attribute**)
What is interacting with **object1**?	(**object2**, **relation**)

evidence in the responses：
（1）referred to as episodic memories [21]
（2）grounded on a specific image

These responses can be harvested from :
  (1) human annotation
  (2) a pre-trained predictive model
need a strategy to generate the best query to ask at the current memory state.

Reinforcement learning (RL) approaches are commonly used to learn such a querying policy.
standard deep RL methods such as DQN [28] have convergence issues in our problem setting with a large discrete action space.
To address this limitation, use a tree expansion method with a greedy scoring function instead.
use supervised learning method to train a query scoring network, which evaluates query candidates at the current state.

query scoring network：
  an MLP model
  followed by two-level hierarchical soft-max for：
    （1） the query types and
    （2）the query objects correspondingly
takes an image-question-memory triplet as **input**
does not take answer vectors as input

no ground-truth labels of the optimal queries at each step,
automatically generate the training samples by Monte-Carlo rollouts.

a rollout procedure of the query tree expansion method.
each node in the tree represents a query candidate.
At each step, maintain a set of nouns seen in question and responses,
branch out queries from this set.

The noun set is initialized by all the noun entities in the question.
This set constrains the width of the search tree, making computation tractable.

During test, the query scoring network computes a score for each terminal node.
The model proposes the next query with the highest score.

3.4. Learning

the core network can be trained end-to-end.
at each step, the query generator makes a hard decision on which query to propose, introducing a non-differentiable operation, yet there exists interdependence between the core network and the query generator.

devise an EM-style training procedure:

(1) freeze the core network while training the query scoring network.
(2) freeze the query scoring network while training the core network.

bootstrap with a uniformly random strategy as the seed query generator, as initially no trained query scoring network.
The initial core network is trained with random rollouts using back-propagation.
In subsequent iterations, the core network is trained with rollouts generated from a trained query generator (i.e., tree expansion + query scoring network) from previous step.
Freezing the core network, we then train the query scoring network.
Train the query scoring network with the image-question-memory triplets as input.
The training set is automatically generated by the core network on Monte-Carlo rollouts.
In each rollout, we add a pair of input and label (i.e., query type and query object) to the training set if the newly added memory flipped previously incorrect predictions to the correct answers.

3.5. Implementation Details

follow the same network setup and the same hyper-parameters as [17].
the core network and the query scoring network have 8,192 hidden units.
use dropout (0.5) after the first layer
ReLU as the non-linearity.
Both networks are trained using SGD with momentum and a base learning rate of 0.01.
perform Monte-Carlo rollouts by the query generator using an #-greedy strategy (Line 7 in Algorithm 1), where # is annealed from 1.0 to 0.1 as the iterative training procedure (N = 5) proceeds.