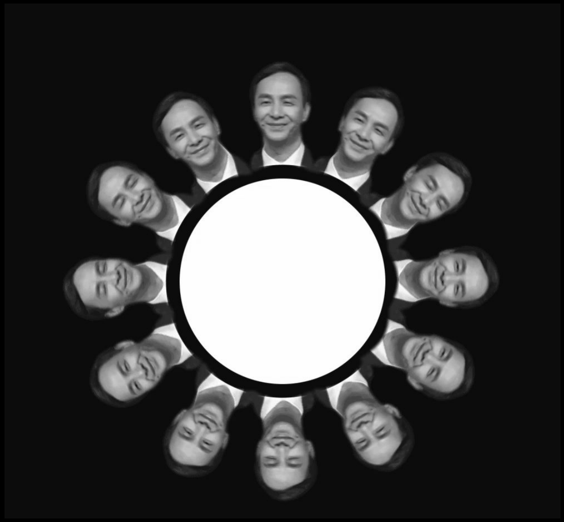
INTRODUCTION TO

VISUAL QUESTION ANSWERING

沈昇勳
Sheng-syun Shen

Outline

- Classical Question Answering
- End-to-End Viausal Question Answering
- Attention Model on Question Answering
- Libraries and Toolkits



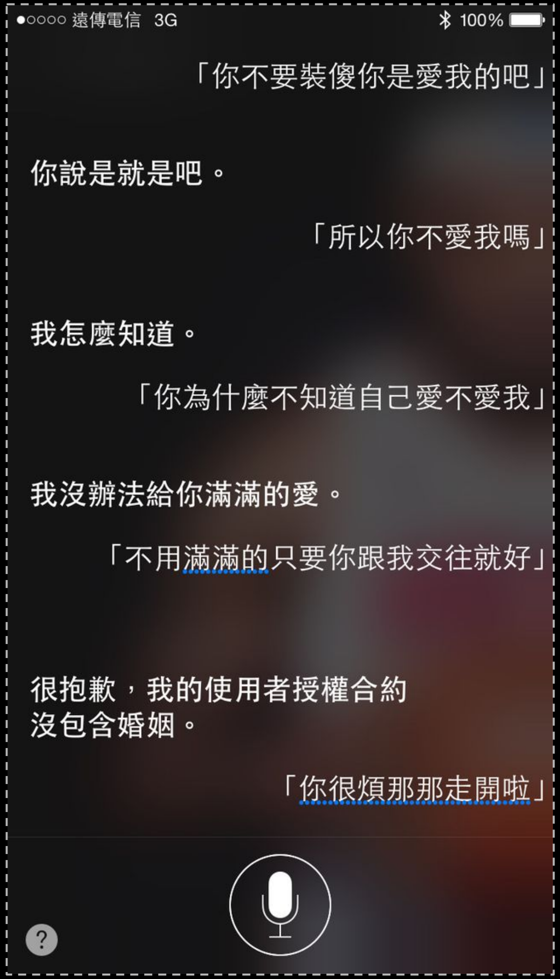
VQA

Classical Question Answering

Question Answering

One of the oldest NLP tasks.

Apple Siri



Types of Questions in QA Sysyem

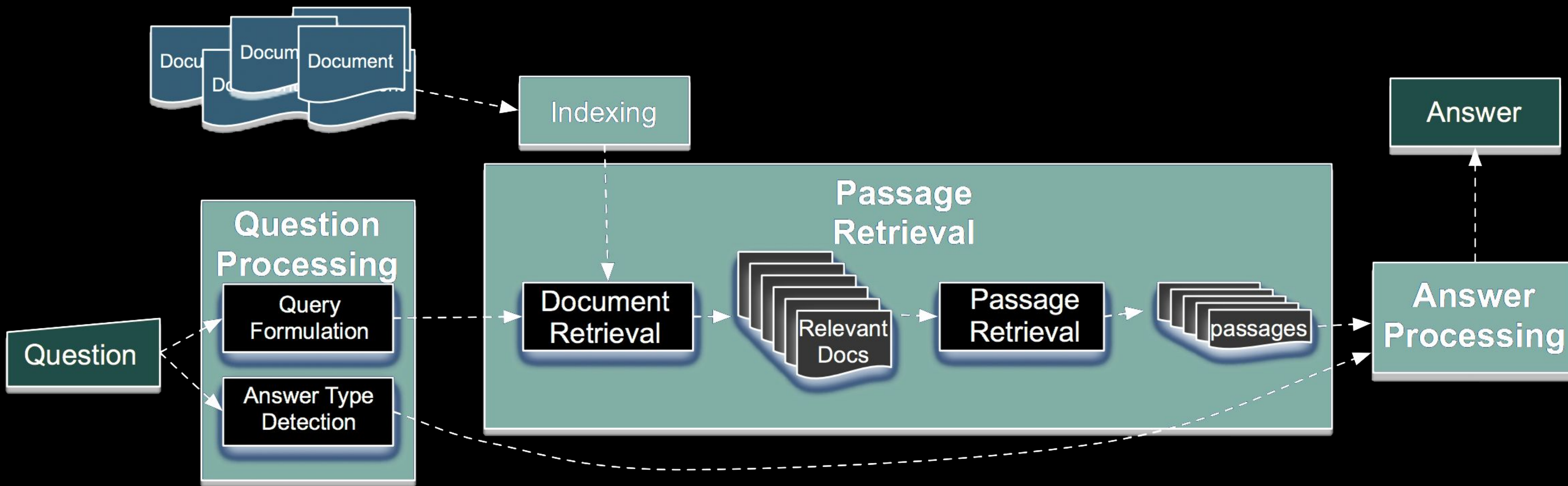
- Factoid questions
  - Where is Apple Computer based ?
  - How many calories are there in two slices of apple pie ?
- Complex (Narrative) questions
  - In children with an acute febrile illness, what is the efficacy of acetaminophen in reducing fever ?

VQA

Approaches for Solving QA

- IR-based approaches (Information Retrieval)
  - TREC; IBM Watson; Google
- Knowledge-based and Hybrid approaches
  - Apple Siri; Wolfram Alpha

IR-based Factoid QA



VQA

IR-based Factoid QA

- Question processing
  - Detect question type, answer type
  - Formulate queries to send to a search engine
- Passage retrieval
  - Retrieve ranked documents
  - Break into suitable passages and rerank
- Answer processing
  - Extract candidate answers
  - Rank candidates

IR-based Factoid QA | Question Processing

- Answer type detection
  Decide the **named entity type** (person, place) of the answer
- Query formulation
  Choose **query keywords** for the IR system
- Question type classification
  Is this a definition question, a math question, a list question

IR-based Factoid QA | Question Processing

Answer type detection : Name entities

- Who founded Virgin Airlines ?
  - PERSON
- What Canadian city has the largest population ?
  - CITY

End-to-End
Viausal Question Answering

Visual QA may contain some sub-problems...

- Object detection
- Image segmentation
- Some Question Answering techniques
  - Question type classification
  - Answer type detection

Is there any banana in the picture ?

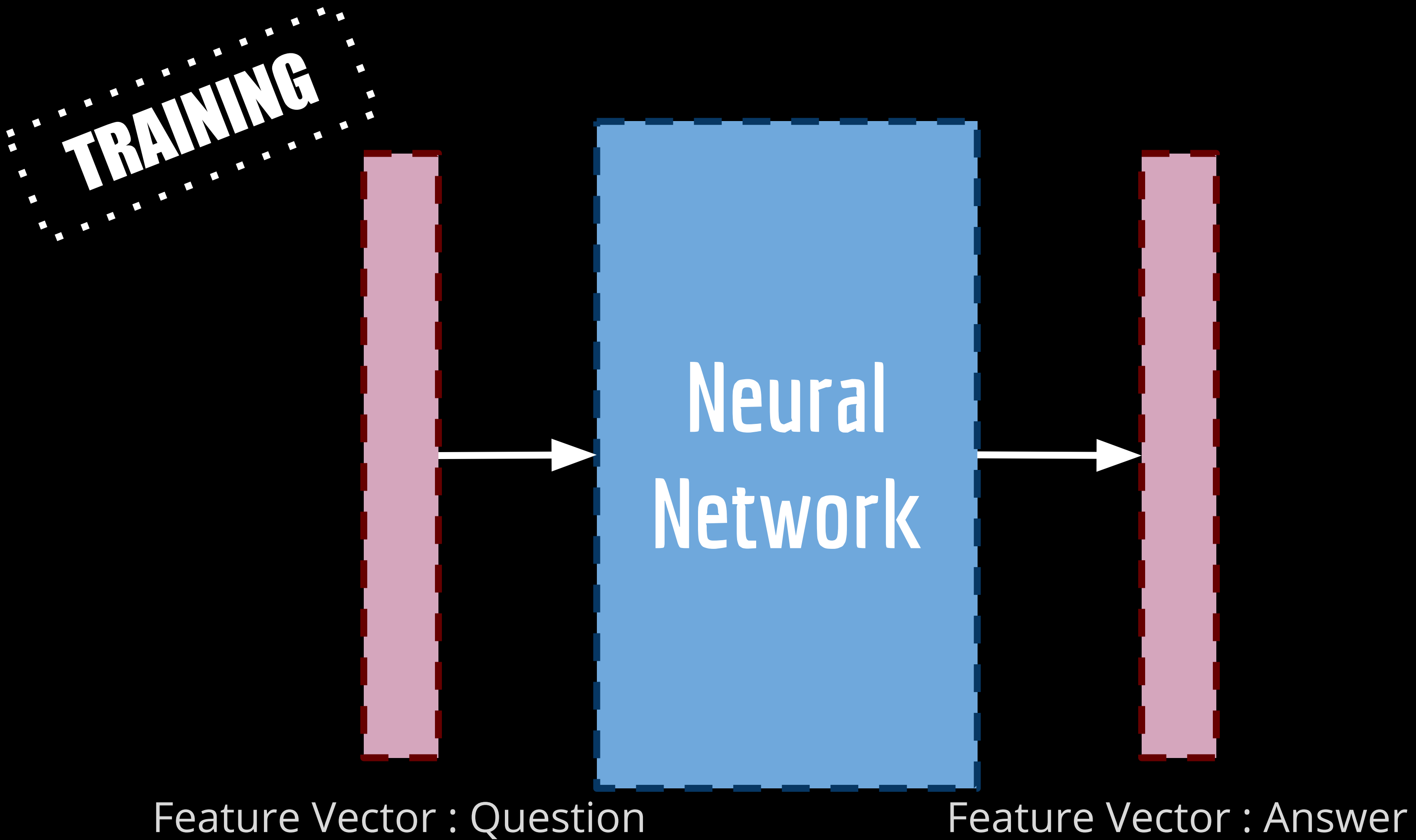
(A) Yes.   (B) No.

VQA

End-to-End Visual QA

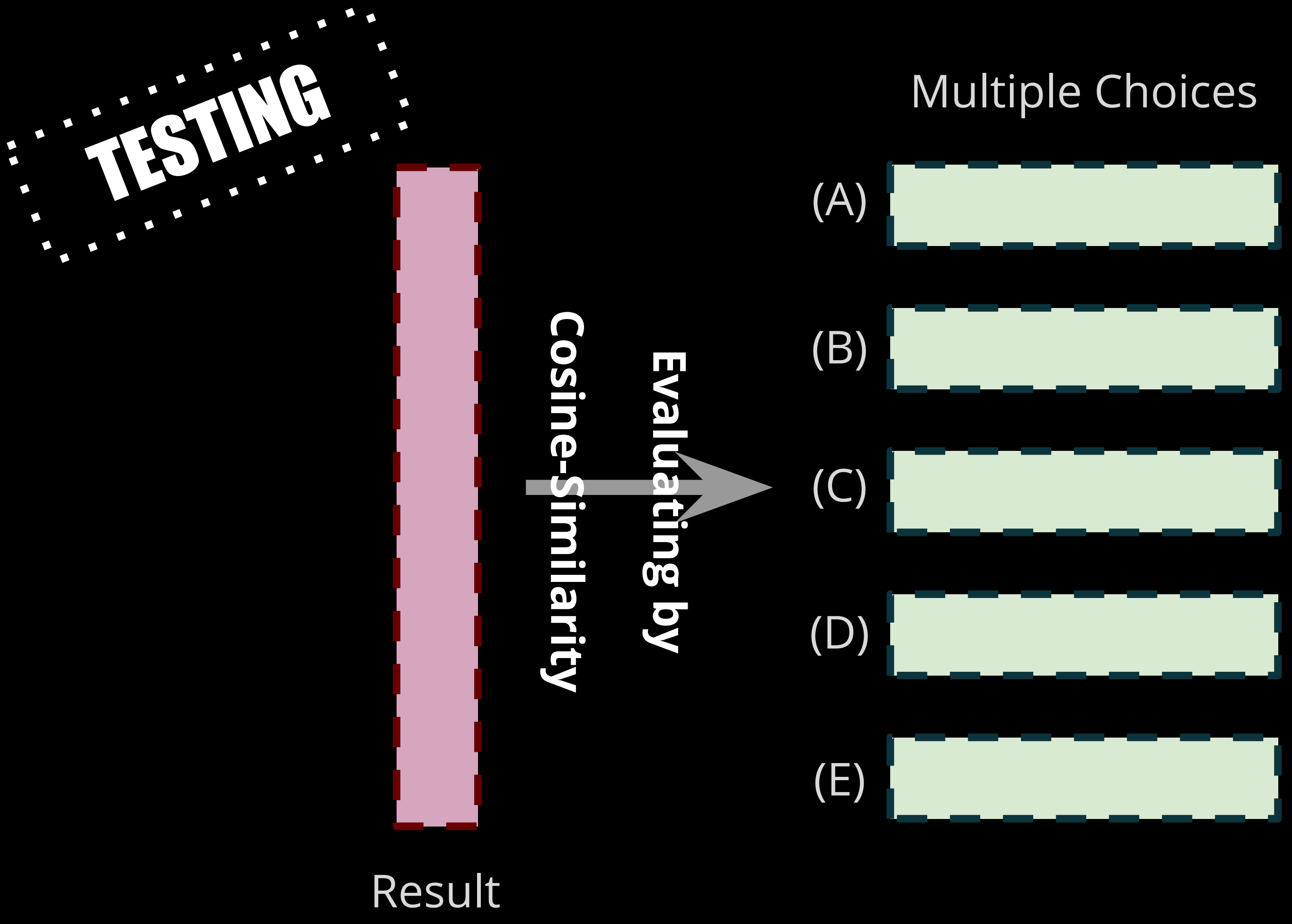
Can directly predict answers according to questions and images

Proposed approach



VQA

Proposed approach



VQA

Extract Feature Vectors | Word Embedding

With a view to understanding sentences or documents, we need to model them in fixed-length vector representation.

Basic Representation Method :

Bag-of-words model / N-hot encoding

- Each document is represented by a set of keywords
- A pre-selected set of index terms can be used to summarize the document contents

Extract Feature Vectors | Word Embedding

Bag-of-words model / N-hot encoding

Definition

- The pre-selected vocabulary $V = \{k_1, \ldots, k_i\}$ is the set of all distinct index terms in the collection
- Examples

$$V = \{John, game, to, likes, watch\}$$

Sentence 1 $S_1 = [1,0,1,2,1]$
John likes to watch movies. Mary likes movies too.

Sentence 2 $S_2 = [1,1,1,1,1]$
John also likes to watch football games.

VQA

Extract Feature Vectors | Word Embedding

Bag-of-words model / N-hot encoding

Property

- Simple and Powerful
- Problem :
    - lose the ordering of the words
    - ignore the semantics of the words

Father = [0 0 0 0 0 1 0 0 … 0 0 0 0]
Mother = [0 0 1 0 0 0 0 0 … 0 0 0 0]

the cosine similarity between these two terms :

= 0 ?!

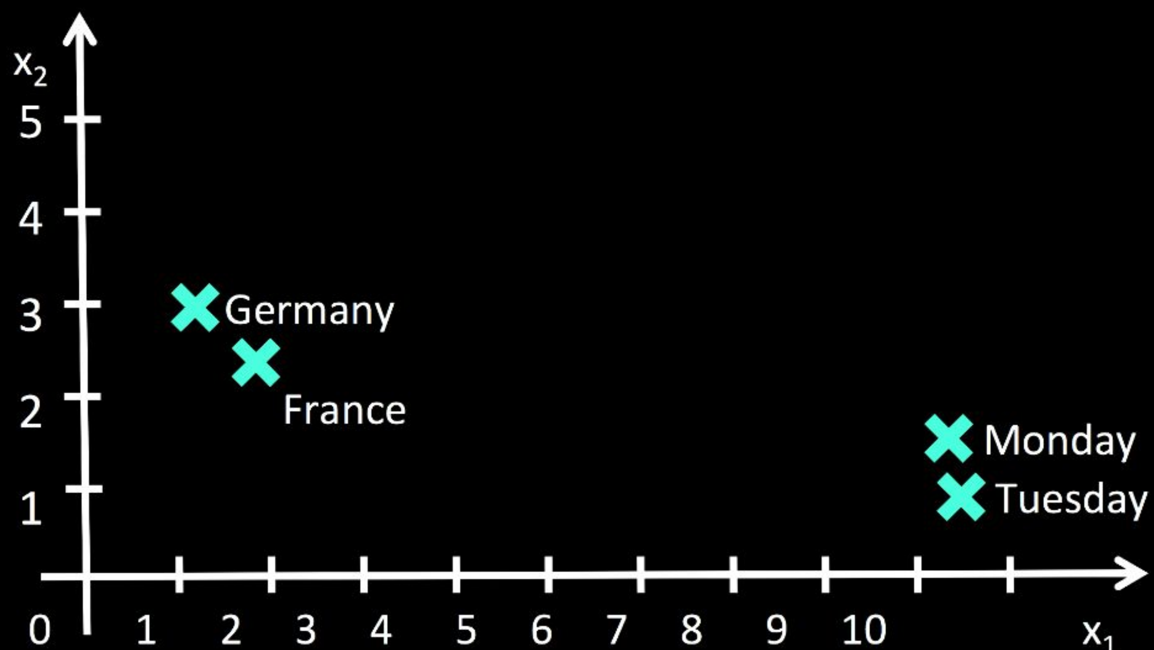
WTF

VQA

Extract Feature Vectors | Word Embedding

While word-embedding can solve these problems :

- Words are represented as a **DENSE, FIX-LENGTH** vector.
- Preserve semantic and syntatic information.

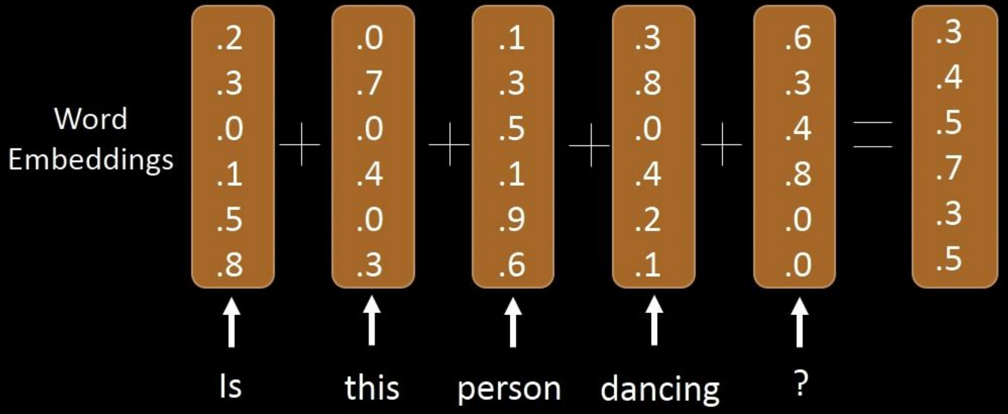


VQA

Extract Feature Vectors | Word Embedding

Using this technique, we can then represent phrases, or sentences by :

- Averaging word vectors

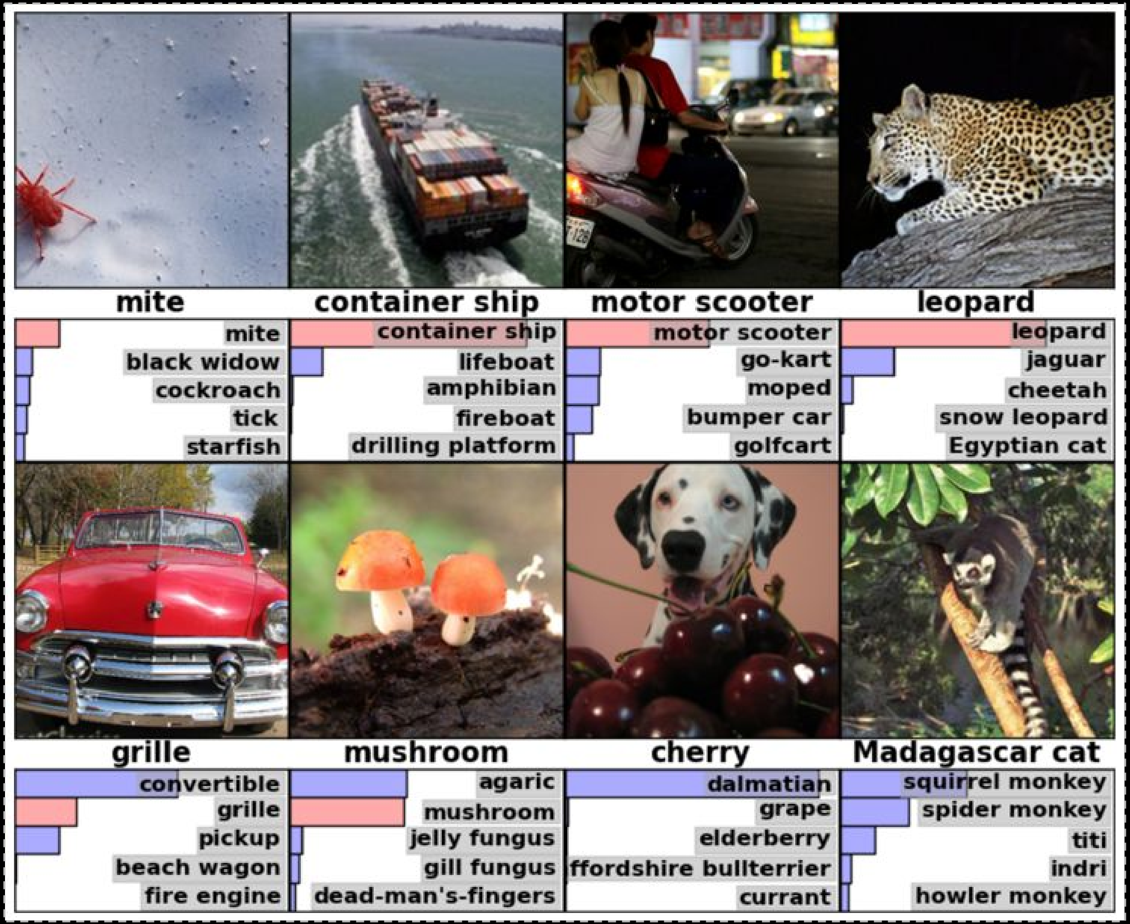


- Adapting sentence-embedding
  https://cs.stanford.edu/~quocle/paragraph_vector.pdf

VQA

Extract Feature Vectors | Image Embedding

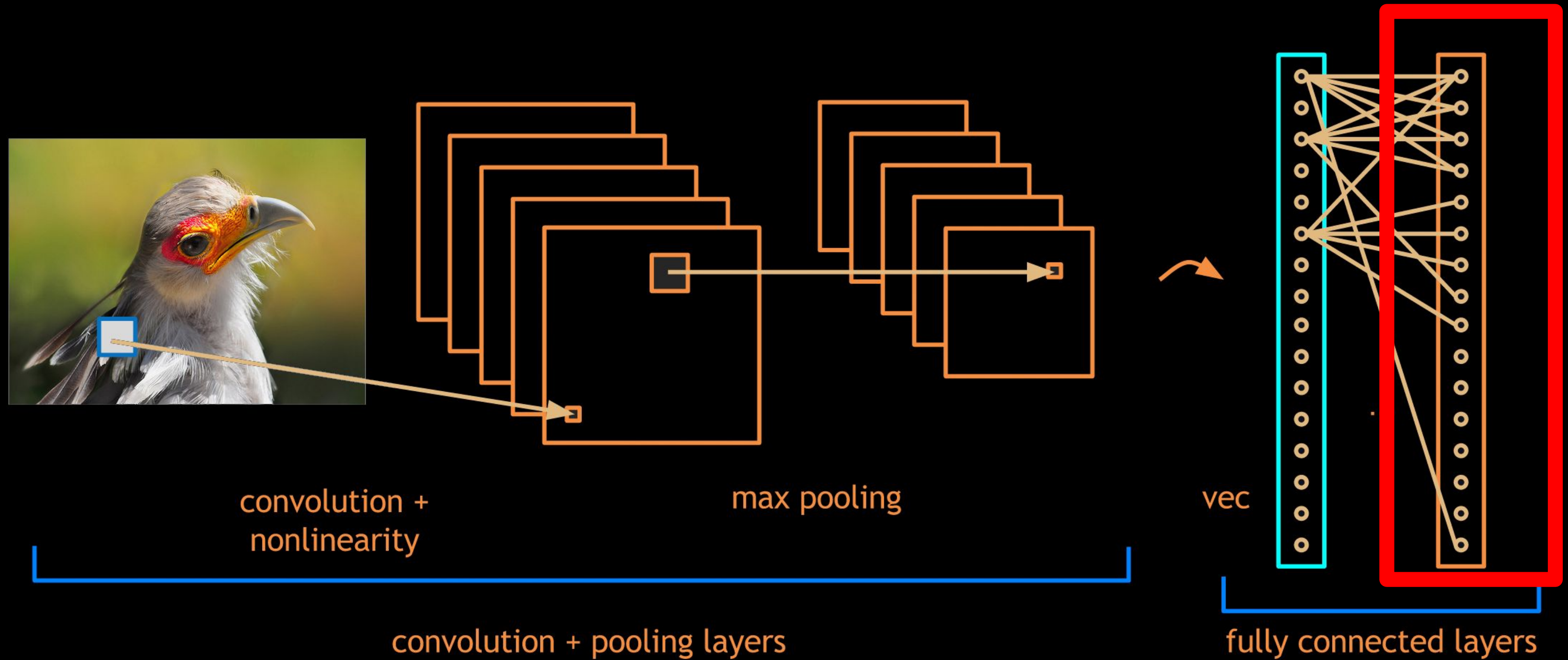
Using a Pre-trained CNN model, we can classify images



VQA

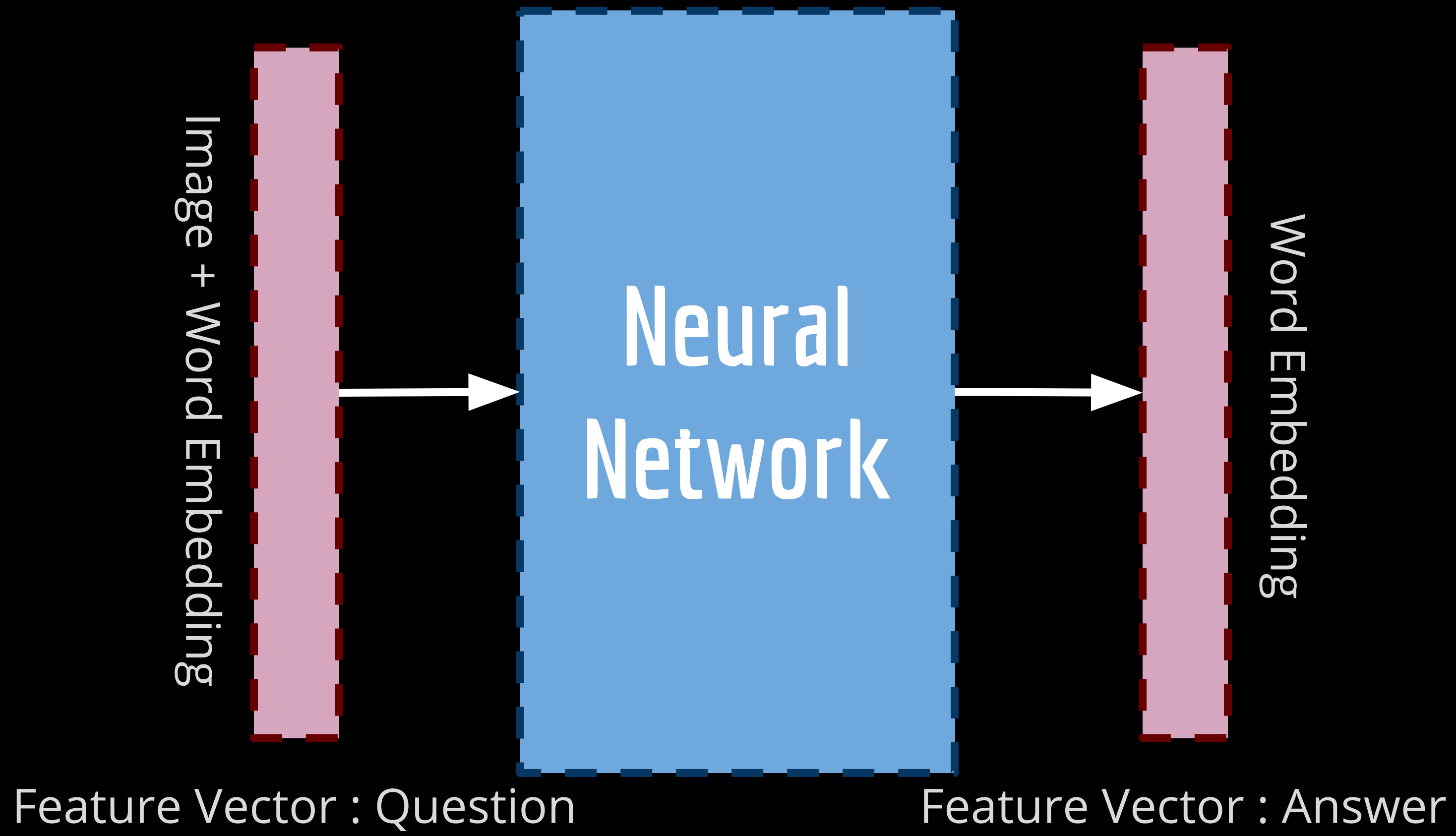
Extract Feature Vectors | Image Embedding

We can also represent images in vector-form by feeding them into the pre-trained CNN models



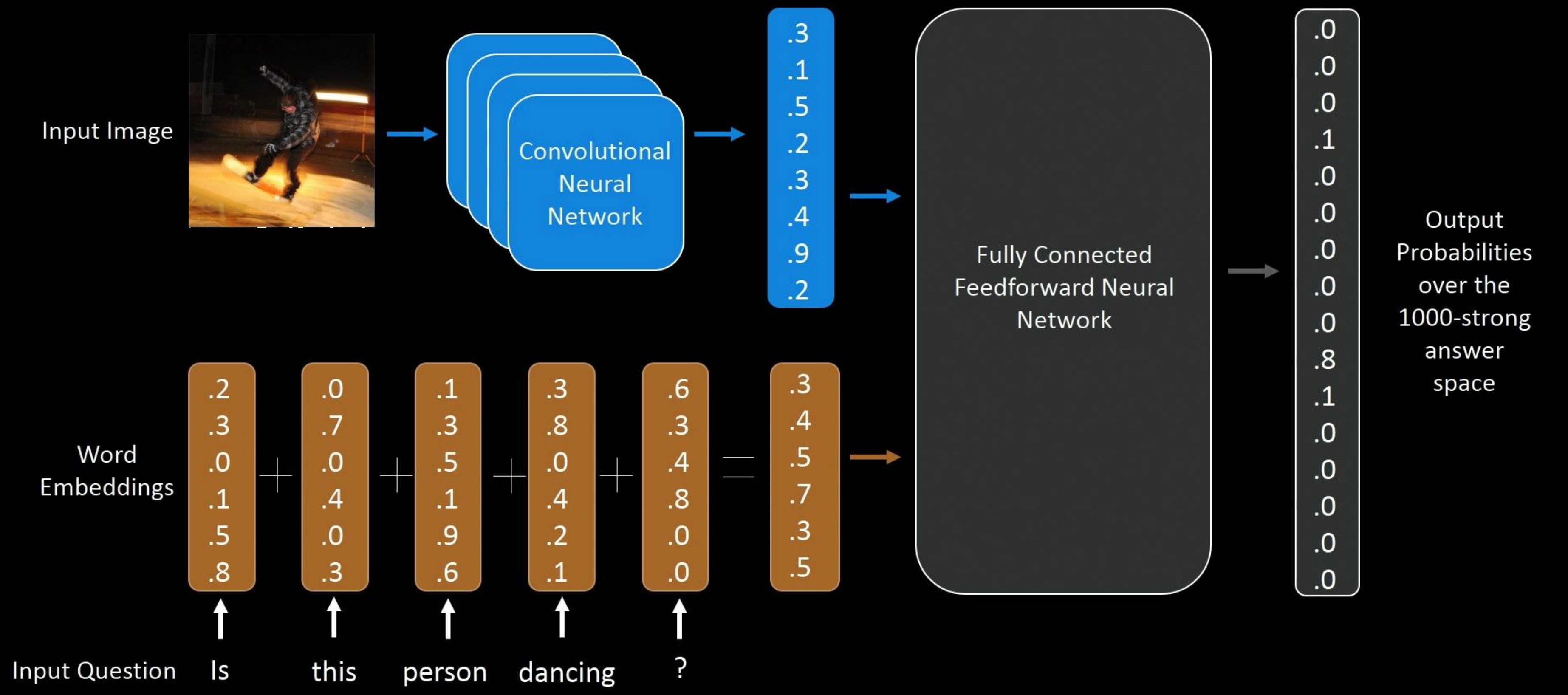
VQA

Proposed approach



VQA

Proposed approach



VQA

Proposed approach

References for implementation :

- https://avisingh599.github.io/deeplearning/visual-qa/
- http://www.cs.toronto.edu/~mren/imageqa/
- https://www.d2.mpi-inf.mpg.de/sites/default/files/iccv15-neural_qa.pdf

VQA

Variations

- **BOW**
  “Blind” model. BOW+logistic regression
- **LSTM**
  Another “Blind” model.
- **IMG**
  CNN feature without question sentences but question type.

Attention Model on Question Answering

Discussion

How to use image information precisely ?

Reference Paper

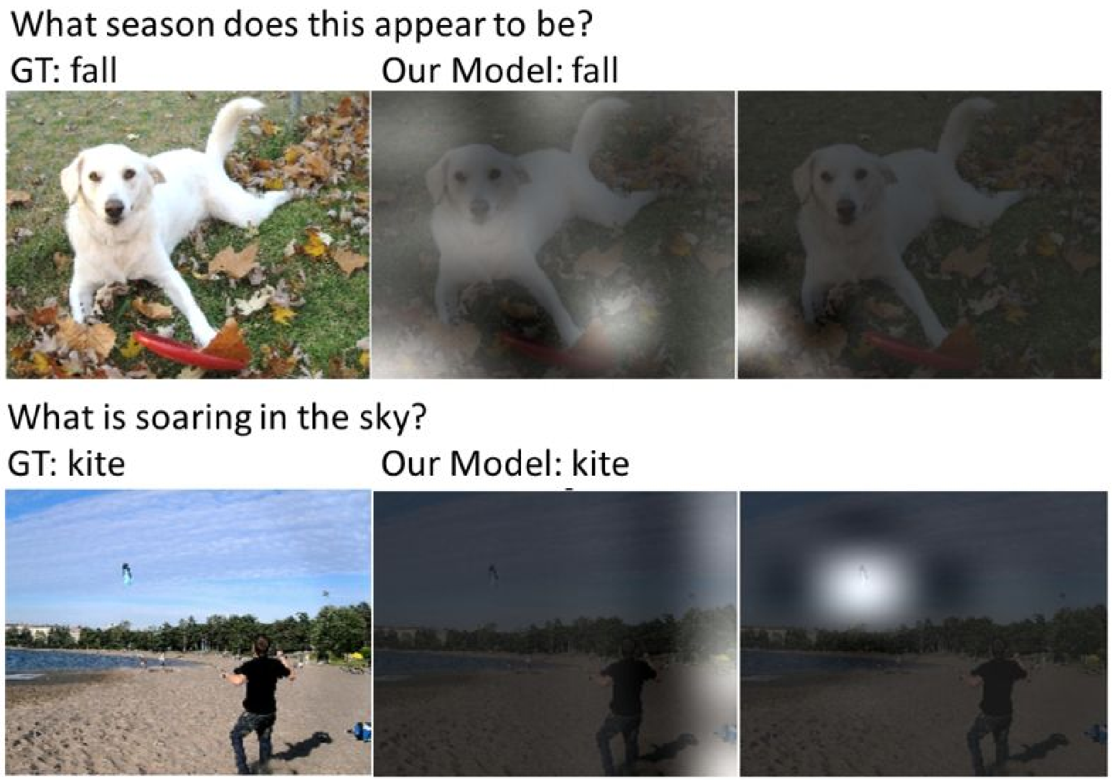
Xu, Huijuan, and Kate Saenko.

UMass Lowell

Ask, Attend and Answer: Exploring Question-Guided Spatial Attention for Visual Question Answering.

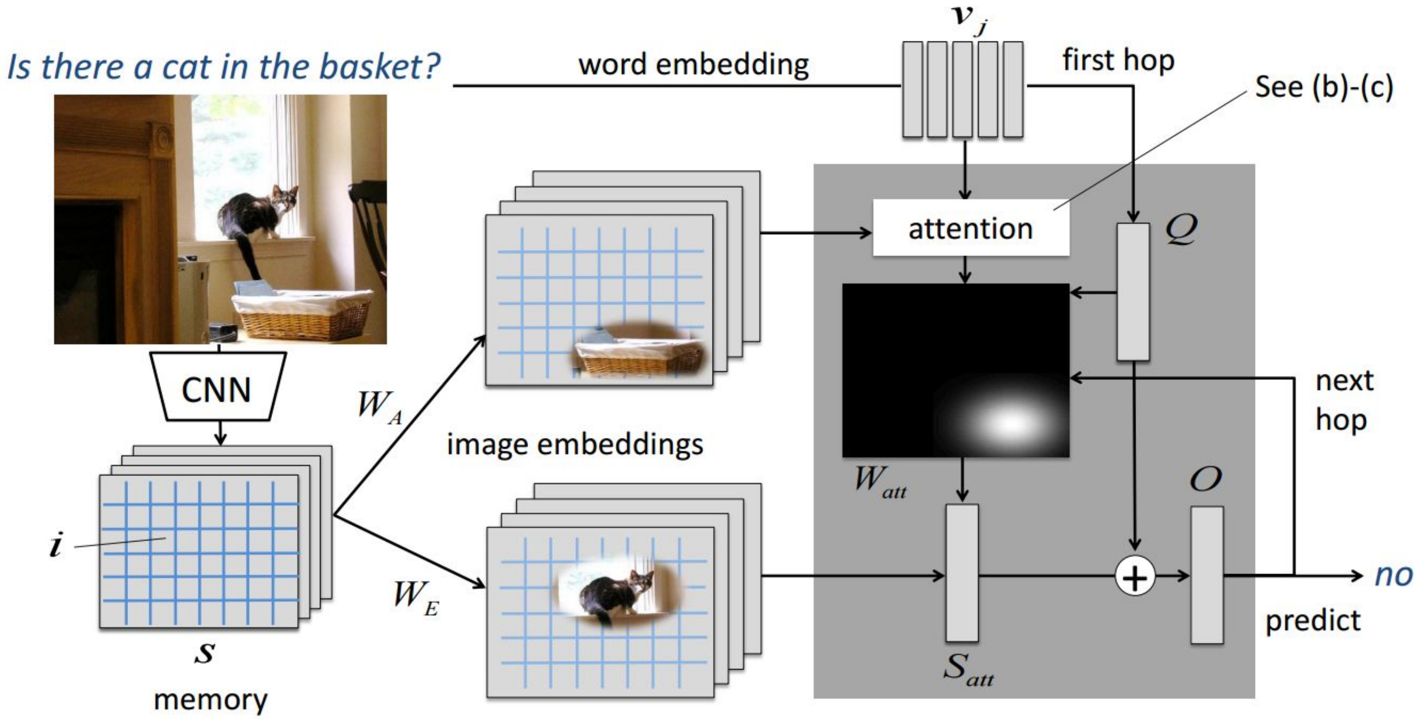
arXiv preprint arXiv:1511.05234 (2015).

Samples in this paper



VQA

Proposed Methodology

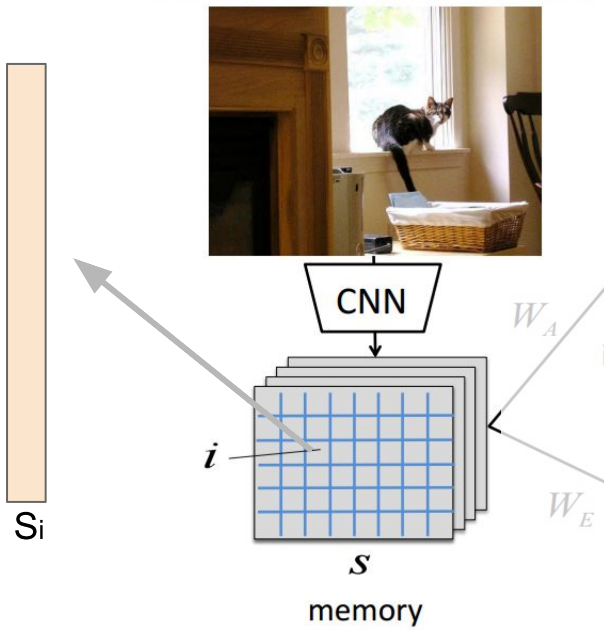


Proposed Methodology

CNN features :

extract the last convolutional layer of GoogLeNet

$$S = \{s_i \mid s_i \in \mathbb{R}^M; i = 1, \cdots, L\}$$



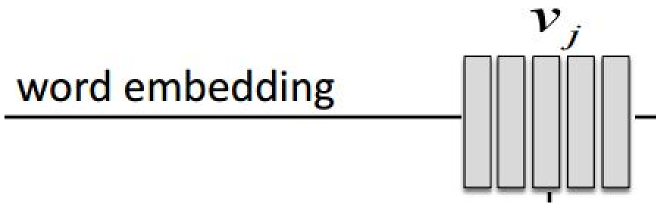
VQA

Proposed Methodology

Text features :

extract the last convolutional layer of GoogLeNet

$$V = \{v_j \mid v_j \in \mathbb{R}^N; j = 1, \cdots, T\}$$



Proposed Methodology | Attention Level

Sentence (Question) Attention

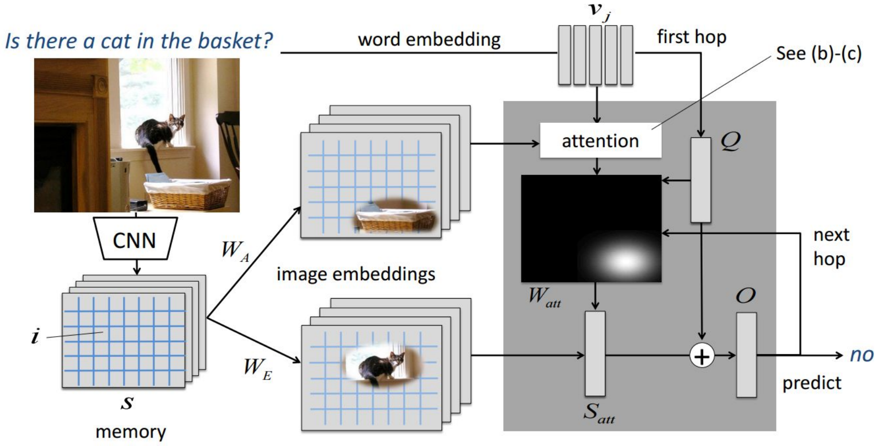
Attention Matrix : W_A

$$C = (S \times W_A) \times Q$$

$$W_{att} = \mathrm{softmax}(C)$$

$$S_{att} = W_{att} \times (S \times W_E)$$

$$P = \mathrm{softmax}(W_P \times (S_{att} + Q) + B_P)$$

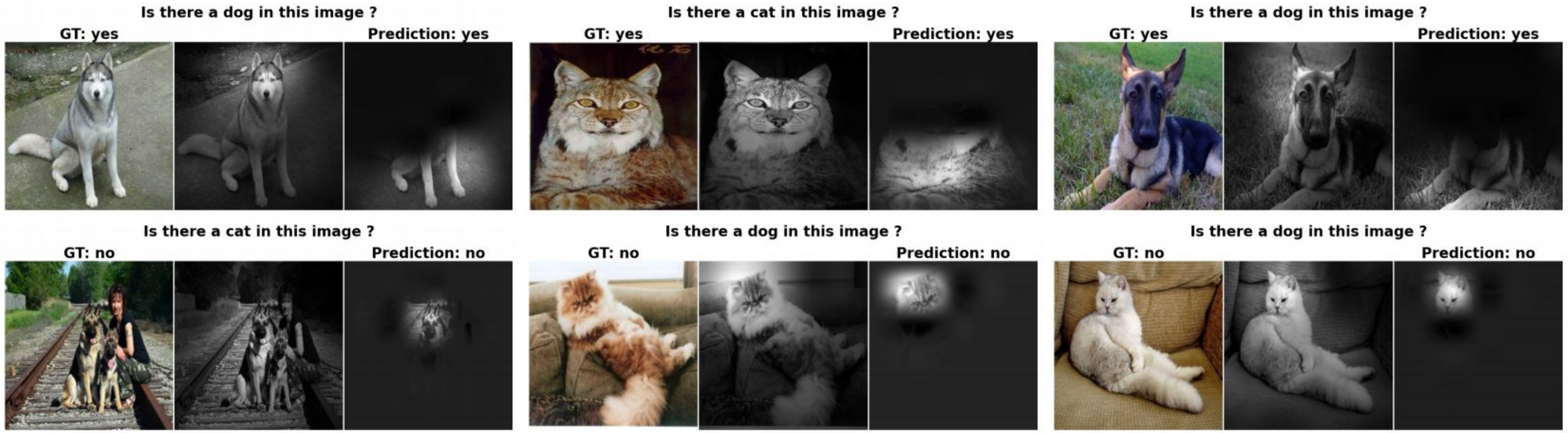


C: R^L, S: $R^{L \times M}$, W_A: $R^{M \times N}$, Q: R^N, W_{att}: R^L, W_E: $R^{M \times N}$

VQA

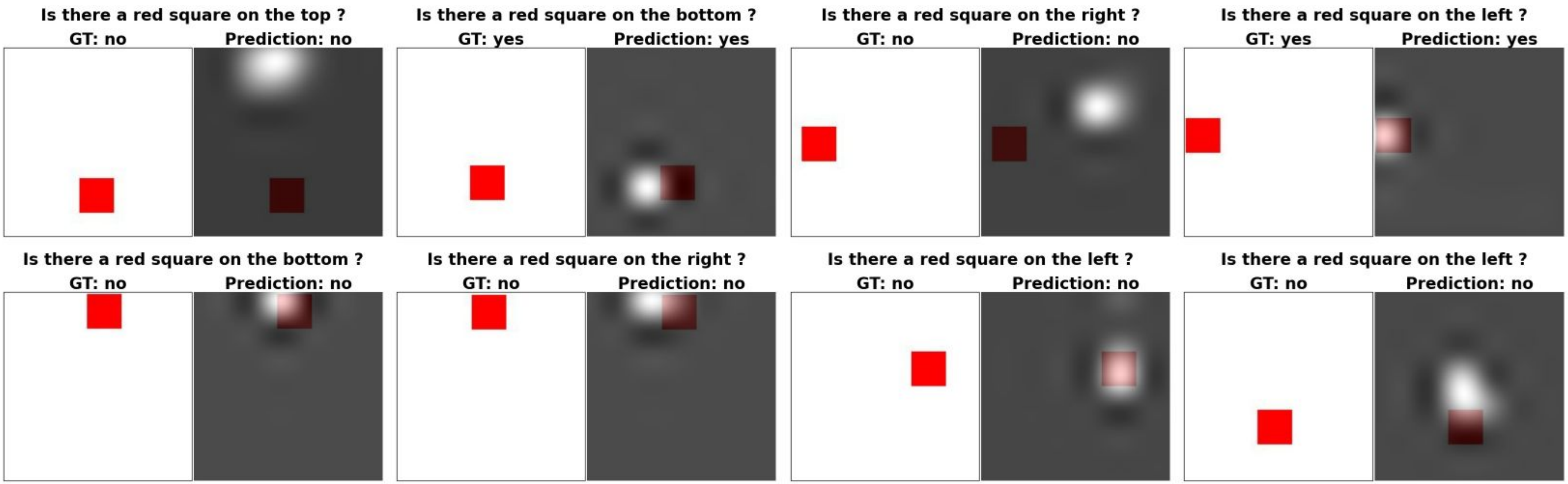
Attention Analysis

Object Presence



Attention Analysis

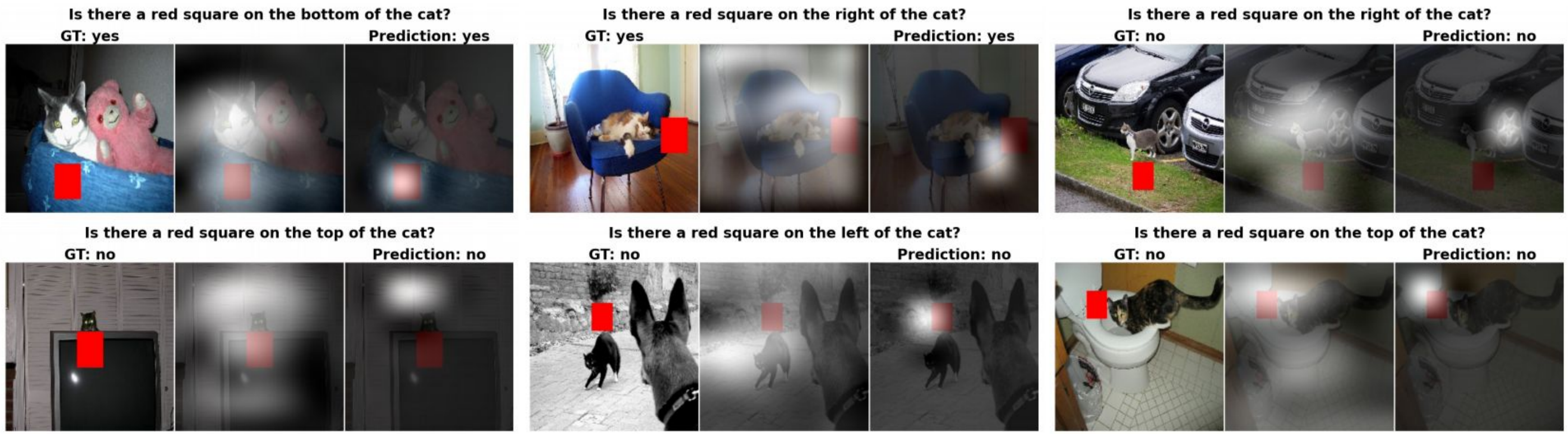
Absolute Position Recognition



With/O : 100% vs 75%

Attention Analysis

Relative Positition Recognition



With/O : 96% vs 75%

Experimental Result

	VQA	DAQUAR	DAQUAR*
Multi-World [17]	-	-	12.73
Neural-Image-QA [18]	51.04	30.64	29.27
Question LSTM [18]	49.73	32.66	32.32
VIS+LSTM [20]	49.54	36.03	34.41
Question BOW [20]	49.67	36.36	32.67
IMG+BOW [20]	53.57	36.03	34.17
Question One-Hop	53.37	36.03	-
Word One-Hop	53.62	36.03	-
Two-Hop	**54.69**	**40.07**	-

Libraries and Toolkits

Word Embedding

- Word2Vec
  https://code.google.com/p/word2vec/
- GloVe
  http://nlp.stanford.edu/projects/glove/
- Sentence2vec
  https://github.com/klb3713/sentence2vec

Image Embedding

An pre-extracted feature set is provided :

http://cs.stanford.edu/people/karpathy/deepimagesent/coco.zip

This is the web page. Hope it works for you :

http://cs.stanford.edu/people/karpathy/deepimagesent/

(It’s about generating image descriptions.)

VQA

Keras

- Website and documentation : http://keras.io/
- Example :

Multilayer Perceptron (MLP):

```
from keras.models import Sequential
from keras.layers.core import Dense, Dropout, Activation
from keras.optimizers import SGD

model = Sequential()
# Dense(64) is a fully-connected layer with 64 hidden units.
# in the first layer, you must specify the expected input data shape:
# here, 20-dimensional vectors.
model.add(Dense(64, input_dim=20, init='uniform'))
model.add(Activation('tanh'))
model.add(Dropout(0.5))
model.add(Dense(64, init='uniform'))
model.add(Activation('tanh'))
model.add(Dropout(0.5))
model.add(Dense(2, init='uniform'))
model.add(Activation('softmax'))

sgd = SGD(lr=0.1, decay=1e-6, momentum=0.9, nesterov=True)
model.compile(loss='mean_squared_error', optimizer=sgd)

model.fit(X_train, y_train, nb_epoch=20, batch_size=16)
score = model.evaluate(X_test, y_test, batch_size=16)
```

Keras

Notification :

If input features are too large for you, you can load them in batch, and apply batch learning as well.

Here are some examples :

https://github.com/avisingh599/visual-qa/blob/master/scripts/trainMLP.py

References

References

- https://web.stanford.edu/class/cs124/lec/qa.pdf

VQA

The End

Thanks for your listening