INTRODUCTION TO VISUAL QUESTION ANSWERING

沈昇勳
Sheng-syun Shen
Outline

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

One of the oldest NLP tasks.

Apple Siri
Types of Questions in QA System

- 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?
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
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 Visual 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.
End-to-End Visual QA

Can directly predict answers according to questions and images
Proposed approach

Neural Network

Feature Vector: Question

Feature Vector: Answer
Proposed approach

Testing

Result

Cosine-Similarity

Evaluating by

Multiple Choices

(A)

(B)

(C)

(D)

(E)
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.
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

\[\text{Father} = [0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ ... \ 0 \ 0 \ 0 \ 0] \]
\[\text{Mother} = [0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ ... \ 0 \ 0 \ 0 \ 0] \]
the cosine similarity between these two terms:

\[= 0 \ ?!\]
Extract Feature Vectors | Word Embedding

While word-embedding can solve these problems:

- Words are represented as a **DENSE, FIX-LENGTH** vector.
- Preserve semantic and syntactic information.
Extract Feature Vectors | Word Embedding

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

- Averaging word vectors
- Adapting sentence-embedding

Using a Pre-trained CNN model, we can classify images.
We can also represent images in vector-form by feeding them into the pre-trained CNN models.
Proposed approach

Image + Word Embedding

Neural Network

Feature Vector: Question

Feature Vector: Answer

Word Embedding
Proposed approach
Proposed approach

References for implementation:

- [https://avisingh599.github.io/deeplearning/visual-qa/](https://avisingh599.github.io/deeplearning/visual-qa/)
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?
Xu, Huijuan, and Kate Saenko. UMass Lowell

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

Samples in this paper

What season does this appear to be?
GT: fall       Our Model: fall

What is soaring in the sky?
GT: kite       Our Model: kite
Proposed Methodology
Proposed Methodology

CNN features:

extract the last convolutional layer of GoogLeNet

\[ S = \{ s_i \mid s_i \in \mathbb{R}^M; i = 1, \ldots, L \} \]
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: $W_A$

\[
C = (S \times W_A) \times Q
\]

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

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

\[
P = \text{softmax}(W_P \times (S_{att} + Q) + B_P)
\]

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

Object Presence

<table>
<thead>
<tr>
<th>Image 1</th>
<th>Image 1</th>
<th>Image 1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GT:</strong> yes</td>
<td><strong>Prediction:</strong> yes</td>
<td><strong>GT:</strong> yes</td>
</tr>
<tr>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image1.png" alt="Image" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Image 2</th>
<th>Image 2</th>
<th>Image 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GT:</strong> yes</td>
<td><strong>Prediction:</strong> yes</td>
<td><strong>GT:</strong> yes</td>
</tr>
<tr>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Image 3</th>
<th>Image 3</th>
<th>Image 3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GT:</strong> yes</td>
<td><strong>Prediction:</strong> yes</td>
<td><strong>GT:</strong> yes</td>
</tr>
<tr>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Image 4</th>
<th>Image 4</th>
<th>Image 4</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GT:</strong> no</td>
<td><strong>Prediction:</strong> no</td>
<td><strong>GT:</strong> yes</td>
</tr>
<tr>
<td><img src="image4.png" alt="Image" /></td>
<td><img src="image4.png" alt="Image" /></td>
<td><img src="image4.png" alt="Image" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Image 5</th>
<th>Image 5</th>
<th>Image 5</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GT:</strong> no</td>
<td><strong>Prediction:</strong> no</td>
<td><strong>GT:</strong> no</td>
</tr>
<tr>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image5.png" alt="Image" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Image 6</th>
<th>Image 6</th>
<th>Image 6</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GT:</strong> no</td>
<td><strong>Prediction:</strong> no</td>
<td><strong>GT:</strong> no</td>
</tr>
<tr>
<td><img src="image6.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
</tr>
</tbody>
</table>
Attention Analysis

Absolute Position Recognition

With/O: 100% vs 75%
Attention Analysis

Relative Position Recognition

With/O : 96% vs 75%
## Experimental Result

<table>
<thead>
<tr>
<th>Method</th>
<th>VQA</th>
<th>DAQUAR</th>
<th>DAQUAR*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-World [17]</td>
<td>-</td>
<td>-</td>
<td>12.73</td>
</tr>
<tr>
<td>Neural-Image-QA [18]</td>
<td>51.04</td>
<td>30.64</td>
<td>29.27</td>
</tr>
<tr>
<td>Question LSTM [18]</td>
<td>49.73</td>
<td>32.66</td>
<td>32.32</td>
</tr>
<tr>
<td>VIS+LSTM [20]</td>
<td>49.54</td>
<td>36.03</td>
<td>34.41</td>
</tr>
<tr>
<td>Question BOW [20]</td>
<td>49.67</td>
<td>36.36</td>
<td>32.67</td>
</tr>
<tr>
<td>IMG+BOW [20]</td>
<td>53.57</td>
<td>36.03</td>
<td>34.17</td>
</tr>
<tr>
<td>Question One-Hop</td>
<td>53.37</td>
<td>36.03</td>
<td>-</td>
</tr>
<tr>
<td>Word One-Hop</td>
<td>53.62</td>
<td>36.03</td>
<td>-</td>
</tr>
<tr>
<td>Two-Hop</td>
<td>54.69</td>
<td>40.07</td>
<td>-</td>
</tr>
</tbody>
</table>
Libraries and Toolkits
Word Embedding

- Word2Vec
  [https://code.google.com/p/word2vec/](https://code.google.com/p/word2vec/)

- GloVe

- Sentence2vec
  [https://github.com/klb3713/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. )
Keras

Website and documentation: [http://keras.io/](http://keras.io/)

Example:

```python
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)
```
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

- 懶得寫了
The End
Thanks for your listening