Visual Question Answering with Various Feature Combinations: Extensions of Visual Question Answering

Jinwoo Choi JINCHOI@VT.EDU
Virginia Tech, 415 Whittemore Hall, Blacksburg, VA 24061 USA

Siddharth Narayanan NSIDDH3@VT.EDU
Virginia Tech, Whittemore Hall, Blacksburg, VA 24061 USA

Abstract

In this work, we present extensions of visual question answering (VQA) task. VQA task is that given an image, a machine is asked to answer a free-form, open-ended, natural-language question about the image. We extend the VQA task to three models: 1) question and answer pair retrieval model, 2) image retrieval model and 3) jeopardy model. Our approach leverages datasets of images and their sentence descriptions to learn about the inter-modal correspondences between language and visual data. We measure a cosine similarity between two modalities based on deep multimodal similarity model (DMSM). Using DMSM, we conduct experiments with different experimental settings comprising of natural images, questions and answers. We look into how accurately different feature combinations generates the predictions.

1. Introduction

Artificial Intelligence (AI) research has recently received significant attention. Among the various problems of AI, image/video captioning which is a combination of Computer Vision, Natural Language Processing and Knowledge Representation and Reasoning has been tackled by plenty of research groups in the past year (Antol et al., 2015; Devlin et al., 2015; Fang et al., 2015; Chen & Lawrence Zitnick, 2015; Karpathy & Fei-Fei, 2015; Vinyals et al., 2015; Donahue et al., 2015). However a significant big gap still exists in the quality of assessment of images as compared to humans.

The goal of our project is to enable machines to understand images along with their corresponding questions and answers and respond to them appropriately. The Motivation behind our problem is to look at all forms of inputs and outputs as opposed to the basic model of a VQA (Antol et al., 2015) task that given an image, a machine is asked to answer a free-form, open-ended, natural-language question about the image. Thus, we extend the VQA task to three different models: 1) question and answer (QA) pair retrieval model, 2) image retrieval model, 3) jeopardy model. The four models in this work are illustrated in the Figure 1.

In the Figure 1, first model is a basic VQA model. Given an Image, ask a question about it, and find the correct answer to the question about the image. In the second model, given an image, we retrieve the corresponding question and answer pair. Third model is an image retrieval model with an input question and answer pair. Given a question and answer pair, we retrieve the most relevant images to the query question and answer pair. The last model is jeopardy model. Given an image and an answer, this model tries to find the question about them. Although there have been several papers on VQA task recently (Antol et al., 2015; Tu et al., 2014; Bigham et al., 2010; Malinowski & Fritz, 2014; Geman et al., 2015), to the best of our knowledge, this is the first research of the VQA extension tasks.

The VQA extensions help us to explore new applications such as enabling the machine to find most appropriate question when given a combination of an image and an answer as an input, or when provided with an image, ask most relevant questions about an image and then answering it correctly. Such a scheme could find use in the original VQA applications like generic object recognition and holistic scene understanding. It could also be useful in narrating information and stories from images, or developing interactive educational applications that ask questions about images.

This paper is organized as follows. In the next section, we illustrate the approach we take to tackle the VQA and three

Virginia Tech CSx824/ECEx424 final project report.

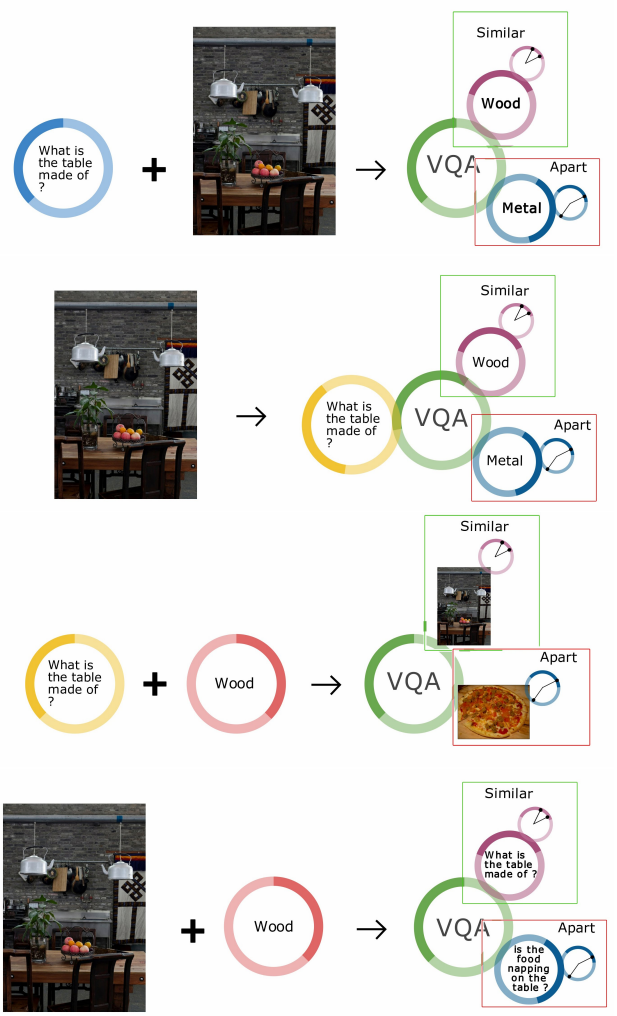

Figure 1. Illustrative examples of the original VQA model and three extension models. First row is the original VQA model. Second row is the QA retrieval model. Third row is the image retrieval model. The last row is the jeopardy model.

extension tasks. After that, we describe the dataset, the implementation details, and show the quantitative and qualitative experimental results of the VQA model and three extension models. Finally, we draw a conclusion of this work and discuss about the future works.

2. Approach

In order to perform a VQA and the VQA extension tasks, we want to measure a similarity between (image & sentence) or (image + sentence & sentence). In this work, we use Deep Multimodal Similarity Model (DMSM) (Fang et al., 2015) to map the input vectors to common semantic space and then measure a cosine similarity between the embedding vectors. In the next subsection, we will briefly introduce the DMSM. We will illustrate the training and prediction pipelines in the following subsections.

2.1. DMSM

The DMSM is a multimodal extension of the unimodal Deep Structured Semantic Model (DSSM) (Huang et al., 2013; Shen et al., 2014). The DSSM is a model to measure a similarity between text queries and documents. DSSM is extended to a multimodal model DMSM by (Fang et al., 2015). The DMSM is a pair of neural networks. At a prediction time, each network takes an image or a text as an input. And then the DMSM maps the input image and the input text to a common semantic space. At the semantic space, the DMSM measure a cosine similarity between an image embedding vector and a text embedding vector. In the DMSM, the cosine similarity R between a query Q, a corresponding embedding vector y_Q, a document D and a corresponding embedding vector y_D is defined as follows.

$$R(Q, D) = \cos(y_Q, y_D) = \frac{y_Q^T y_D}{||y_Q|| ||y_D||} \tag{1}$$

For the training, we can define the posterior probability of a document D in the set of candidate documents $\mathbb{D}$ as follows.

$$p(D|Q) = \frac{\exp(\gamma R(Q, D))}{\sum_{D' \in \mathbb{D}} \exp(\gamma R(Q, D'))} \tag{2}$$

And the objective function with a model parameter Λ for the DMSM training is defined as follows:

$$L(\Lambda) = -\log \prod_{(Q, D^+)} p(D^+|Q) \tag{3}$$

Here, D^+ is a positive document in the candidate documents set $\mathbb{D}$. Using the objective function 3 and back-propagation, the DMSM updates the weight matrices of each layer. The DMSM training and prediction stages are illustrated in the Figure 2.

2.2. VQA Extensions Training Pipeline

Using DMSM, VQA training pipeline in this work can be depicted as the Figure 3. First row represents the basic VQA model pipeline, second row represents the QA retrieval model pipeline and the image retrieval model pipeline. Since the QA retrieval model and the image retrieval model have same input features, the only difference between them is a fact that what is a query and what is a document. Third row represents the jeopardy model training pipeline.

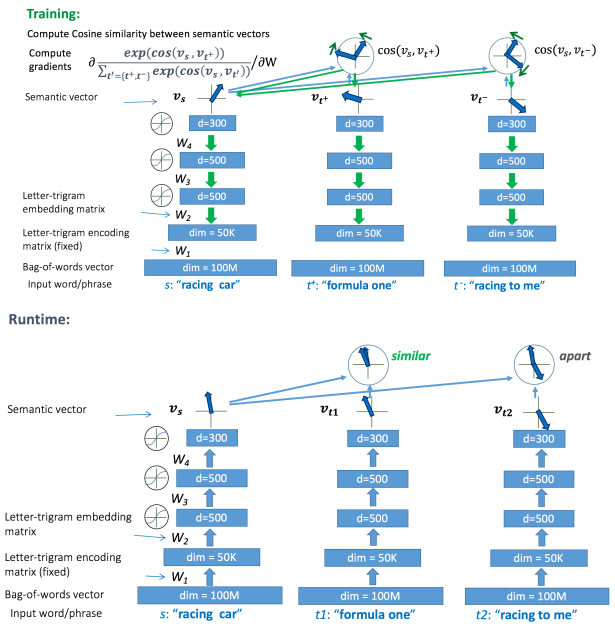

Figure 2. An illustration of DMSM training and prediction stages.

For the VQA model, we extract features from images and concatenate the image features with question features. (Image+question) features and answer features are fed into the query network and the document network of DMSM respectively to train the model.

For the QA retrieval model, extracted image features are fed into the query network of DMSM. And the concatenated (question+answer) features are fed into the document network of DMSM. For the image retrieval model, the concatenated (question+answer) features are fed into the query network and extracted image features are fed into the document network of DMSM respectively.

For the jeopardy model, we concatenate image features with answer features. The concatenated (image+answer) features and question features are fed into the query network and the document network of DMSM respectively to train the DMSM.

In this work, an open source deep learning framework Caffe (Jia et al., 2014) is used to extract the image features which are activations from VGGNET (Simonyan & Zisserman, 2014). For question and answer features, we represent the features as bag-of-word representations of letter-trigram count vectors following (Huang et al., 2013).

2.3. VQA Extensions Prediction Pipeline

A VQA model prediction pipeline is depicted in the Figure 4. For the VQA model, the prediction in DMSM returns cosine similarity scores between a query (an image + question feature) and documents (candidate answer features) ranging from -1 to 1. We then select the maximum scored answer among the 18 multiple choices per each question.

The prediction pipelines for the other three models are straightforward modifications of the VQA model prediction pipeline. We concatenate the features according to the model, and feed the two inputs to DMSM. The only difference between the original VQA model and the three extension models is a score space. Since the three extension models are retrieval models, we should compute the similarities between a given query and entire documents. Then we re-rank the scores in a descending order. Finally we can take the top-K most similar documents as the query results.

3. Experiment

In this section, we first describe the dataset we used, the implementations issues in detail, followed by the experimental settings of DMSM parameters. We will show the quantitative and qualitative results of the VQA model and the three extensions models.

3.1. Dataset

In this work, we used VQA dataset (Antol et al., 2015). This dataset contains 82,783 training images and 40,504 validation images of newly released Microsoft COCO dataset (Chen et al., 2015; Lin et al., 2014). Furthermore, the VQA dataset contains approximately 3 ground truth question-answer pairs per each training/validation image. The VQA dataset provides two modalities for answering the questions: (1) open-answer and (2) multiple-choice. In this work, we only conducted experiments on multiple-choice answers for the VQA model. We used the training images and corresponding training questions and answers to train our models. We used validation images and corresponding validation questions and answers to get predictions and to calculate the accuracies. Please see (Antol et al., 2015; Chen et al., 2015) for more details about the dataset.

3.2. Implementation Details

We have conducted experiments using different combinations of features. We first generate feature sets of the required combination, align them and concatenate them to get a sparse vector representation. The open source deep learning framework Caffe (Jia et al., 2014) is used to extract the image features which are activations from VGGNET (Simonyan & Zisserman, 2014). We then feed both features to the DMSM to update weight matrices. Different input combinations of image, question and answer are used in this work. In the test time, we measure the similarity

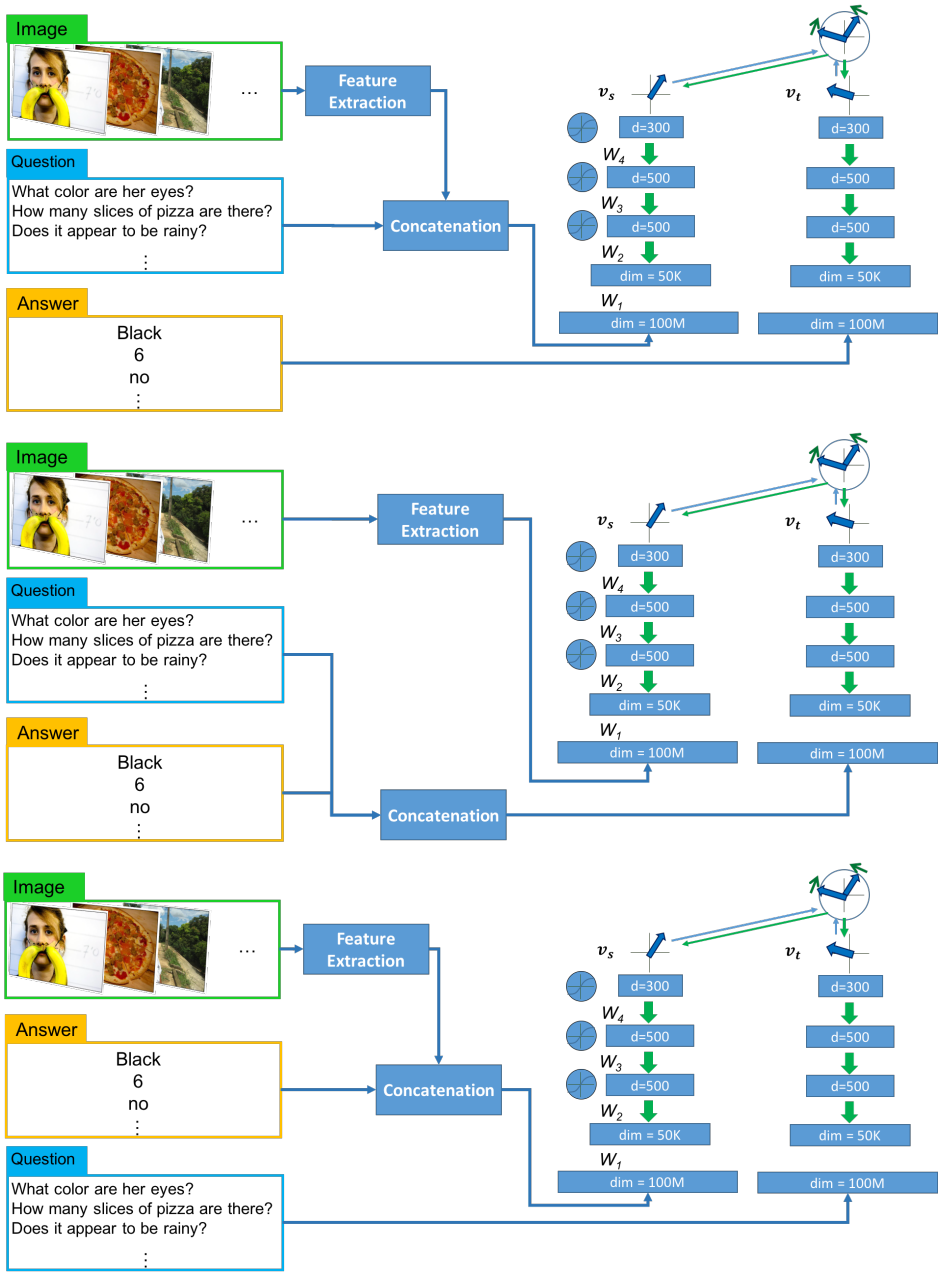

Figure 3. Training pipeline of the VQA model (first row), the QA retrieval model, the image retrieval model (second row) and the jeopardy model (third row).

between the two inputs by a cosine similarity between their embedding vectors.

For example, we compute the embeddings for a given image and use the multimodal cosine similarity score to find the nearest question+answer embedding for the image. In other words we demonstrate the ability to inquire what questions can be asked about the image and ability to get the answers of the automatically generated questions.

There were hurdles that we faced during the implementation of training and testing, primarily concerning the system performance. Given the large size of the dataset, we had decided to use the GPU systems in the CVMLP lab to run the training and testing. However, the performance

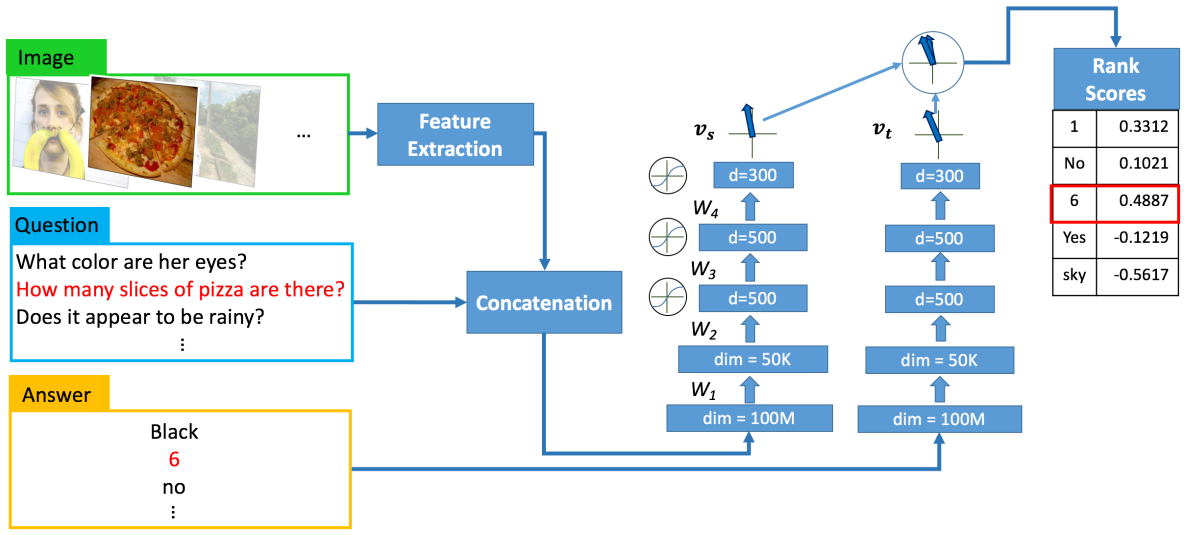

Figure 4. Prediction pipeline of the VQA model.

on the GPU systems using the Torch library[1] was still very slow, taking 2–3 hours per each epoch. We then explored the possibility of running our implementation using the DMSM C# reference code[2] on a regular CPU configuration, since we did not have access to a GPU-equipped Windows machine. We observed that the results were relatively better in this case, now taking 70–100 minutes per each epoch. Therefore, for 100 epochs, we need approximately 5–7 days to train a model. We also observed that both methods of implementation yielded almost equivalent results, the C# code had a training loss of 670456.9 for the first epoch of the original VQA model, while the Torch code had a loss of 699201.3 for the first epoch.

Some effort was also required to align the dataset for the concatenation to generate features. For example, in the question answer validation pair file, there was an anomaly where the question was "What brand of motorcycle is this ?" and the corresponding answer was "?". Thus we had to filter out and align the data so as to avoid such instances in the training.

3.3. Experimental Setting

The four experiments conducted in this work is summarized in the Table 1. We used the same input parameters for all of the four experimental models. Batch size was set to 1024. Maximum number of epochs were set to 100. Learning rate was set to 0.02 and we used tanh as the activation function both for query and document networks. We used MMI as an objective criterion. We ran our experiments without GPU, because we did not have any GPU-equipped Windows machine.

Table 1. Experimental model list

NUMBER	DESCRIPTION	ABBREVIATION
1	ORIGINAL VQA MODEL	IQ,A
2	QA RETRIEVAL MODEL	I,QA
3	IMAGE RETRIEVAL MODEL	QA,I
4	JEOPARDY MODEL	IA,Q

3.4. Quantitative Results

For the original VQA model, we used accuracy (the number of correct question-answer pairs divided by the total number of question-answer pairs) as an evaluation metric. The accuracy for the multiple-choice VQA model can be computed as follows:

1) We obtain all similarity scores among 18 multiple-choice answers to a given question. 2) We select an answer with the maximum score of 18 multiple choices. 3) If the selected answer is equal to the ground truth answer, it is considered as a correct answer. 4) The accuracy is then defined as (# of correct question-answer pairs)/(total # of question-answer pairs).

[1]In this work, we used the DSSM/DMSM Torch implementation written by Jiasen Lu: https://github.com/jiasenlu/CDSSM.git

[2]In this work, we used the DSSM/DMSM source code released by Microsoft Research: http://research.microsoft.com/en-us/downloads/731572aa-98e4-4c50-b99d-ae3f0c9562b9/

Table 2. The original VQA model accuracy (multiple-choice answer) compared with obvious baselines.

RANDOM CHOICE	ALL YES	VQA MODEL
0.29	26.80	46.41

Quantitative results for the original VQA task are shown in the Table 2. We compare our VQA model with two baseline methods. First baseline is that we select an answer randomly from 18 multiple-choice for each question. Second baseline is we always answer "yes" which is the most common answer in the dataset. All the accuracy values of compared baseline are measured by ourselves. As shown in the table, using image features with question features, we can answer more accurate than baseline methods.

For the three extension models, we also used accuracy as an evaluation metric. However, definition of accuracy is different. For the QA retrieval, image retrieval and jeopardy models, accuracy is (the number of successful queries)/(the total number of query-document pairs). Here, we treat a query as successful if the query is matched to the corresponding document within K-top retrieved documents according to descending order of the similarities.

We measured accuracies with various K values and the accuracies are summarized in the Table 3. As expected, our model accuracies increase with increasing K values. We also compared the proposed three extension models with random baseline. Our model accuracies 39–77 times outperform the random baseline. For the cases that K values are ranging from 1 to 200, someone can argue that the accuracies are not high enough. However, we only treated the exact match of ground truth pairs as a successful query. There can be a reasonable pair even if it is not a ground truth pair. For example, second row of the Figure 7, all of the top-3 retrieved images contain elephants and they are not swimming. Thus the effective accuracy values of the extension models would be higher than the values in the Table 3.

3.5. Qualitative Results

Some example results of the original VQA model are shown in the Figure 5. As shown in the top row, for some cases machines can answer the questions given an image correctly. Machines can answer the abstract concept questions (third example in the Figure 5) as well as just simple questions (fourth example in the Figure 5). However, there are also some failure cases as shown in the bottom row. For the first example of the failure cases, machine answers that sky is completely clear of cloud even though there are plenty of clouds in the image. But these failure cases implies that there is a room for improvement of the VQA accuracy.

QA pair retrieval example results are depicted in the Figure 6. Given a query image, the machine retrieves the corresponding QA pairs and re-ranks the pairs according to the similarity scores in a descending order. We remarked the correct answer rank in the figure. Even though the correct answer rank is not very high, retrieved answers are relevant to the query image. In the first example, all top-3 ranked QA pairs contain "table" and "fruit" which are contained in the query image. In the third example, two out of 3-top ranked QA pairs contain "cat" while the other misclassified a cat in the image as a dog.

Some examples of the image retrieval results are shown in the Figure 7. Given a query QA pair, the machine retrieves the corresponding images and re-ranks the images according to the similarity scores. We remarked the correct answer rank in the figure. Similar to the QA retrieval results, top-3 retrieved images are reasonably relevant to the query QA pair. For example, the second row in the Figure 7 has "Are the elephants swimming" and "no" as a QA pair. All top-3 retrieved images contain elephants and they are not swimming. The third row in the Figure 7 has "Where is this person cooking this meal" and "oven" as a QA pair. All top-3 retrieved images contain cooking scene. And two of them contain the oven.

Jeopardy model example results are depicted in the Figure 8. Given an image and an answer, the machine tries to find what is the question. We remarked the correct answer rank in the figure. This model also retrieves the relevant questions reasonably. In the first example, the query image is a restroom image and a query answer is "tile". All top-3 retrieved questions contain "floor" or "ceiling". In the third example, the query image contains a scene about a man is skateboarding and the answer is "skateboarding". All top-3 retrieved questions have a form of "What is someone doing". But the top rank question is "what is the fireman doing" which is definitely not correct. This is because of the blurry part of the image. Blur of the lights might have yielded the confusions to image feature extraction module.

4. Conclusion and Future Works

We have implemented a VQA and three VQA extension schemes using the dataset of images and their sentence descriptions to learn about the inter-modal correspondences between language and visual data. Inputs are taken in the form of either images, questions and answers. Experimental results show some promising results of the VQA extension models.

As a future work, we can exploit the three extension models to improve the original VQA model. For example,

Table 3. QA Retrieval/Image Retrieval/Jeopardy model accuracies with various K values compared with a random baseline.

MODEL	RANDOM	K=1	K=10	K=100	K=200	K=1000	K=2000
QA RETRIEVAL	0.004	0.31	2.51	16.37	26.14	59.41	72.54
IMAGE RETRIEVAL	0.004	0.23	1.92	12.59	20.00	47.74	59.85
JEOPARDY	0.01	0.39	2.62	13.32	20.46	44.83	57.27

Figure 5. VQA model example results: top row shows successful cases, bottom row shows failure cases.

Figure 6. QA pair retrieval model example results.

given an image, a question and multiple choice answers, we can calculate the IA,Q embeddings and get the similarities. And then we can use these similarities to help the original VQA task. Applying transfer learning technique to the three extension models using the weights of the original VQA model can be another future work. Transfer learning may reduce the training time of the three extension models and might increase the model accuracies.

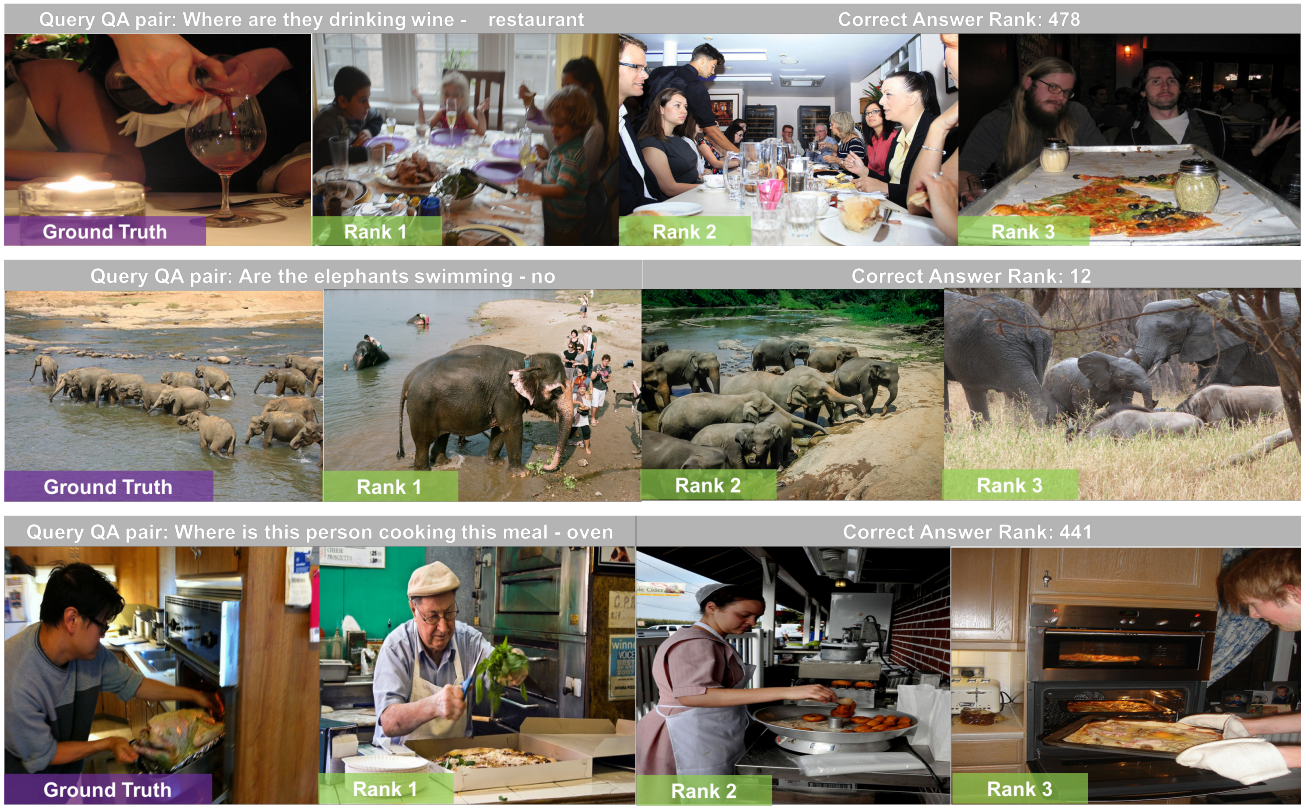

Figure 7. Image retrieval model example results.

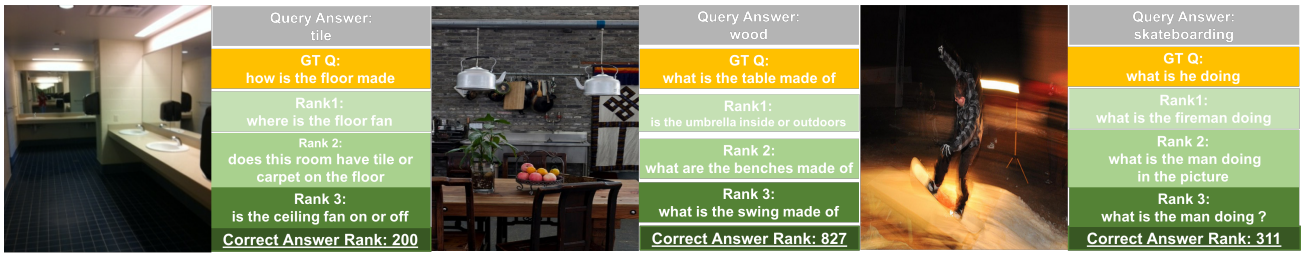

Figure 8. Jeopardy model example results.

References

Antol, Stanislaw, Agrawal, Aishwarya, Lu, Jiasen, Mitchell, Margaret, Batra, Dhruv, Zitnick, C. Lawrence, and Parikh, Devi. Vqa: Visual question answering. In *International Conference on Computer Vision (ICCV)*, 2015.

Bigham, Jeffrey P, Jayant, Chandrika, Ji, Hanjie, Little, Greg, Miller, Andrew, Miller, Robert C, Tatarowicz, Aubrey, White, Brandyn, White, Samuel, and Yeh, Tom. Vizwiz: nearly real-time answers to visual questions. In *Proceedings of the 2010 International Cross Disciplinary Conference on Web Accessibility (W4A)*, pp. 24. ACM, 2010.

Chen, Xinlei and Lawrence Zitnick, C. Mind's eye: A recurrent visual representation for image caption generation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

Chen, Xinlei, Fang, Hao, Lin, Tsung-Yi, Vedantam, Ramakrishna, Gupta, Saurabh, Dollar, Piotr, and Zitnick, C Lawrence. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

Devlin, Jacob, Gupta, Saurabh, Girshick, Ross, Mitchell, Margaret, and Zitnick, C Lawrence. Exploring nearest neighbor approaches for image captioning. *arXiv preprint arXiv:1505.04467*, 2015.

Donahue, Jeffrey, Anne Hendricks, Lisa, Guadarrama, Sergio, Rohrbach, Marcus, Venugopalan, Subhashini, Saenko, Kate, and Darrell, Trevor. Long-term recurrent convolutional networks for visual recognition and description. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

Fang, Hao, Gupta, Saurabh, Iandola, Forrest, Srivastava, Rupesh K., Deng, Li, Dollar, Piotr, Gao, Jianfeng, He, Xiaodong, Mitchell, Margaret, Platt, John C., Lawrence Zitnick, C., and Zweig, Geoffrey. From captions to visual concepts and back. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

Geman, Donald, Geman, Stuart, Hallonquist, Neil, and Younes, Laurent. Visual turing test for computer vision systems. *Proceedings of the National Academy of Sciences*, pp. 201422953, 2015.

Huang, Po-Sen, He, Xiaodong, Gao, Jianfeng, Deng, Li, Acero, Alex, and Heck, Larry. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pp. 2333–2338. ACM, 2013.

Jia, Yangqing, Shelhamer, Evan, Donahue, Jeff, Karayev, Sergey, Long, Jonathan, Girshick, Ross, Guadarrama, Sergio, and Darrell, Trevor. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

Karpathy, Andrej and Fei-Fei, Li. Deep visual-semantic alignments for generating image descriptions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

Lin, Tsung-Yi, Maire, Michael, Belongie, Serge, Hays, James, Perona, Pietro, Ramanan, Deva, Dollár, Piotr, and Zitnick, C Lawrence. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014*, pp. 740–755. Springer, 2014.

Malinowski, Mateusz and Fritz, Mario. A multi-world approach to question answering about real-world scenes based on uncertain input. In *Advances in Neural Information Processing Systems*, pp. 1682–1690, 2014.

Shen, Yelong, He, Xiaodong, Gao, Jianfeng, Deng, Li, and Mesnil, Gregoire. A latent semantic model with convolutional-pooling structure for information retrieval. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pp. 101–110. ACM, 2014.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

Tu, Kewei, Meng, Meng, Lee, Mun Wai, Choe, Tae Eun, and Zhu, Song-Chun. Joint video and text parsing for understanding events and answering queries. *MultiMedia, IEEE*, 21(2):42–70, 2014.

Vinyals, Oriol, Toshev, Alexander, Bengio, Samy, and Erhan, Dumitru. Show and tell: A neural image caption generator. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.