Visual Question Answering

Mohan Bhambhani

Topics in Deep Learning, 2017

Outline

Introduction

The Task

- What does the sign say?

Introduction

The Task

- What does the sign say? stop

Introduction

The Task

- What does the sign say? stop
- What shape is this sign?

Introduction

The Task

- What does the sign say? stop
- What shape is this sign? octagon

Introduction

The Task

- What does the sign say? stop
- What shape is this sign? octagon

- A VQA system takes as input an image and a natural language question about the image and produces a natural language answer as the output.

Outline

COCA-QA

- **Images** - MS-COCO dataset.
- **Questions** - From image descriptions.
- **Answers** - From image descriptions.

COCA-QA

- **Images** - MS-COCO dataset.
- **Questions** - From image descriptions.
- **Answers** - From image descriptions.
- Types of questions:
  - Object: Questions regarding objects in the image (69%).
  - Number: counting based questions (7%).
  - Color: Questions asking color of objects (16%)
  - Location: Questions with places, scenes as the answer (6%)

COCA-QA

- **Images** - MS-COCO dataset.
- **Questions** - From image descriptions.
- **Answers** - From image descriptions.
- Types of questions:
  - Object: Questions regarding objects in the image (69%).
  - Number: counting based questions (7%).
  - Color: Questions asking color of objects (16%)
  - Location: Questions with places, scenes as the answer (6%)
- Example: Using the image caption A boy is playing Frisbee, create the question What is the boy playing?, with frisbee as the answer.

COCA-QA

Dataset details:

Table 1: COCO-QA question type break-down

CATEGORY	TRAIN	%	TEST	%
OBJECT	54992	69.84%	27206	69.85%
NUMBER	5885	7.47%	2755	7.07%
COLOR	13059	16.59%	6509	16.71%
LOCATION	4800	6.10%	2478	6.36%
TOTAL	78736	100.00%	38948	100.00%

COCA-QA

- A side-effect of the automatic conversion of captions is a high repetition rate of the questions.
- Among the 38,948 questions of the test set, 9,072 (23.29%) of them also appear as training questions.
- Top 1000 frequent answers cover 100% of the test answers.

COCA-QA

- A side-effect of the automatic conversion of captions is a high repetition rate of the questions.
- Among the 38,948 questions of the test set, 9,072 (23.29%) of them also appear as training questions.
- Top 1000 frequent answers cover 100% of the test answers.

- Also, due to automatic conversion from captions there are many grammatical and semantic errors.

COCO-QA: What does an intersection show on one side and two double-decker buses and a third vehicle,?
Ground Truth: Building

VQA

- **Images** - Real images from MS-COCO and abstract cartoon scenes.
- The abstract scenes were created manually to represent realistic situations, using a drag-and-drop interface.
- Scenes are made from over 100 different objects, 30 different animal models, and 20 human cartoon models.

VQA

- **Images** - Real images from MS-COCO and abstract cartoon scenes.
- The abstract scenes were created manually to represent realistic situations, using a drag-and-drop interface.
- Scenes are made from over 100 different objects, 30 different animal models, and 20 human cartoon models.
- **Questions** - AMT to generate questions.
- **Answers** - AMT. Separate pool of workers to make answers to the questions.

VQA

- **Images** - Real images from MS-COCO and abstract cartoon scenes.
- The abstract scenes were created manually to represent realistic situations, using a drag-and-drop interface.
- Scenes are made from over 100 different objects, 30 different animal models, and 20 human cartoon models.
- **Questions** - AMT to generate questions.
- **Answers** - AMT. Separate pool of workers to make answers to the questions.
- 3 questions per image, with 10 answers per question.

VQA

- **Images** - Real images from MS-COCO and abstract cartoon scenes.
- The abstract scenes were created manually to represent realistic situations, using a drag-and-drop interface.
- Scenes are made from over 100 different objects, 30 different animal models, and 20 human cartoon models.
- **Questions** - AMT to generate questions.
- **Answers** - AMT. Separate pool of workers to make answers to the questions.
- 3 questions per image, with 10 answers per question.
- Available in both open-ended and multiple choice setting (18 options per question containing the Correct Answer(1), Plausible Answers(3), Popular Answers(10), Random Answers(4))

VQA

Dataset details:

- Real data has about 120k training and 80k test images, and about 600k questions.

VQA

Dataset details:

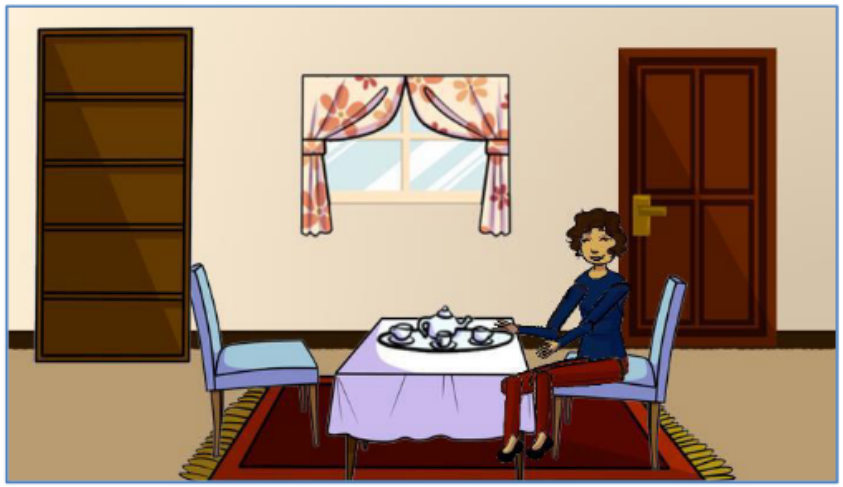
- Real data has about 120k training and 80k test images, and about 600k questions.
- 50k abstract scenes were generated. There are about 150k QA pairs.

VQA

Dataset details:

- Real data has about 120k training and 80k test images, and about 600k questions.
- 50k abstract scenes were generated. There are about 150k QA pairs.

What type of trees are here?	palm palm palm	ash oak pine
Is the skateboard airborne?	yes yes yes	no yes yes



What is the woman doing?	sitting sitting sitting	reading reading watching tv
Who is having tea?	lady woman woman	woman woman women

VQA

- Most of the questions can be accurately answered without using the image due to language biases.

VQA

- Most of the questions can be accurately answered without using the image due to language biases.
- The dataset also contains many opinion-seeking questions that do not have a single objective answer.

VQA

- Most of the questions can be accurately answered without using the image due to language biases.
- The dataset also contains many opinion-seeking questions that do not have a single objective answer.
- Bias in the dataset: 'yes/no' answers span about 38% of all questions, and almost 59% of them are answered with 'yes.'

VQA

- Most of the questions can be accurately answered without using the image due to language biases.
- The dataset also contains many opinion-seeking questions that do not have a single objective answer.
- Bias in the dataset: 'yes/no' answers span about 38% of all questions, and almost 59% of them are answered with 'yes.'
- In VQA-real, the corresponding top-1000 answers cover as much as 80% of the test set answers.

Visual Genome

- **Images:** 108k images that occur in both YFCC100M and MS-COCO images.

Visual Genome

- **Images:** 108k images that occur in both YFCC100M and MS-COCO images.
- **Questions:** AMT.

Visual Genome

- **Images:** 108k images that occur in both YFCC100M and MS-COCO images.
- **Questions:** AMT.
- **Answers:** AMT.

Visual Genome

- **Images:** 108k images that occur in both YFCC100M and MS-COCO images.
- **Questions:** AMT.
- **Answers:** AMT.
- Three rules to make questions:
  - Start the questions with one of the "seven W" (who, what, where, when, why, how and which)
  - Avoid ambiguous and speculative questions
  - Relate the question to the image such that it is clearly answerable if and only if the image is shown.

Visual Genome

- **Images:** 108k images that occur in both YFCC100M and MS-COCO images.
- **Questions:** AMT.
- **Answers:** AMT.
- Three rules to make questions:
  - Start the questions with one of the "seven W" (who, what, where, when, why, how and which)
  - Avoid ambiguous and speculative questions
  - Relate the question to the image such that it is clearly answerable if and only if the image is shown.
- Question collection modes:
  - In free-form method, annotators were free to ask any question about an image.
  - In region-based QA, annotators must write a QA pair based on a given region in image.

Visual Genome

- **Images:** 108k images that occur in both YFCC100M and MS-COCO images.
- **Questions:** AMT.
- **Answers:** AMT.
- Three rules to make questions:
  - Start the questions with one of the "seven W" (who, what, where, when, why, how and which)
  - Avoid ambiguous and speculative questions
  - Relate the question to the image such that it is clearly answerable if and only if the image is shown.
- Question collection modes:
  - In free-form method, annotators were free to ask any question about an image.
  - In region-based QA, annotators must write a QA pair based on a given region in image.
- Dataset also contains scene-graph annotations (contains object, relationships and attributes), region graphs (object boundaries).

Visual Genome

Example:

Q: What is the colour of the fire hydrant in the image ?
A: Yellow

Q: What is the colour of the shirt the man is wearing ?
A: Blue

Scene graph

Visual Genome

Example:

Q: What is the woman doing ?

A: Sitting

Visual Genome

- Over 1.7 million questions. On an average 17 per image.

Visual Genome

- Over 1.7 million questions. On an average 17 per image.
- The Top-1000 frequent answers only cover 65% of all answers in the dataset.

Visual Genome

- Over 1.7 million questions. On an average 17 per image.
- The Top-1000 frequent answers only cover 65% of all answers in the dataset.
- Visual Genome has no binary (yes/no) questions.

Visual7w

- Subset of the Visual Genome that contains additional annotations.

Visual7w

- Subset of the Visual Genome that contains additional annotations.
- Also adds a multiple-choice setting, each question being provided with 4 candidate answers.

Visual7w

- Subset of the Visual Genome that contains additional annotations.
- Also adds a multiple-choice setting, each question being provided with 4 candidate answers.
- Additionally, it has 'pointing' questions. To answer these, the algorithm must select the correct bounding box among alternatives.

Visual7w

- Subset of the Visual Genome that contains additional annotations.
- Also adds a multiple-choice setting, each question being provided with 4 candidate answers.
- Additionally, it has 'pointing' questions. To answer these, the algorithm must select the correct bounding box among alternatives.
- All the objects mentioned in the questions are visually grounded.

Visual7w

- Subset of the Visual Genome that contains additional annotations.
- Also adds a multiple-choice setting, each question being provided with 4 candidate answers.
- Additionally, it has 'pointing' questions. To answer these, the algorithm must select the correct bounding box among alternatives.
- All the objects mentioned in the questions are visually grounded.

Q: Which doughnut has multicolored sprinkles?

Outline

VQA model general architecture

Image Feature Extraction

common choices include:

- Output from the penultimate layer of a pre-trained CNN
- Local features from the convolutional feature maps generated by a pre-trained CNN
- CNN features extracted region proposals
- ...

What animals are these?

Question Feature Extraction

common choices include:

- Bag of words encoding
- LSTM/GRU language models
- Skip-thoughts encoder
- Natural language parser
- ...

Algorithm

common choices include:

- Concatenation
- Elementwise product/sum
- Bilinear pooling
- Attentive models
- Bayesian models
- Compositional models
- ...

CLASSIFIER

Girraffe

Elephant

Cat

Playing

Yes

2

Baseline

Image features: VGGNet
Question features: Bag of words / LSTM
Combining algorithm: Simple mechanisms like concatenation, element-wise multiplication, or element-wise addition

Baseline

Image features:

Baseline

Image features:

- **I:** The activations from the last hidden layer of VGGNet are used as 4096-dim image embedding.

Baseline

Image features:

- **I:** The activations from the last hidden layer of VGGNet are used as 4096-dim image embedding.
- **norm I:** These are l_2 normalized activations from the last hidden layer of VGGNet.

Baseline

Image features:

- **I:** The activations from the last hidden layer of VGGNet are used as 4096-dim image embedding.
- **norm I:** These are l_2 normalized activations from the last hidden layer of VGGNet.

Question features:

Baseline

Image features:

- **I:** The activations from the last hidden layer of VGGNet are used as 4096-dim image embedding.
- **norm I:** These are l_2 normalized activations from the last hidden layer of VGGNet.

Question features:

- **Bag-of-Words BoW Q:** The top 1,000 words in the questions are used to create a bag-of-words representation. The top 10 first, second, and third words of the questions are also added. Thus, 1,030-dim embedding.

Baseline

Image features:

- **I:** The activations from the last hidden layer of VGGNet are used as 4096-dim image embedding.
- **norm I:** These are l_2 normalized activations from the last hidden layer of VGGNet.

Question features:

- **Bag-of-Words BoW Q:** The top 1,000 words in the questions are used to create a bag-of-words representation. The top 10 first, second, and third words of the questions are also added. Thus, 1,030-dim embedding.
- **LSTM Q:** LSTM with one hidden layer is used to obtain 1024-dim embedding for the question. Concatenate cell state and hidden state.

Baseline

Image features:

- **I:** The activations from the last hidden layer of VGGNet are used as 4096-dim image embedding.
- **norm I:** These are l_2 normalized activations from the last hidden layer of VGGNet.

Question features:

- **Bag-of-Words BoW Q:** The top 1,000 words in the questions are used to create a bag-of-words representation. The top 10 first, second, and third words of the questions are also added. Thus, 1,030-dim embedding.
- **LSTM Q:** LSTM with one hidden layer is used to obtain 1024-dim embedding for the question. Concatenate cell state and hidden state.
- **deeper LSTM Q:** An LSTM with two hidden layers is used to obtain 2048-dim embedding for the question.

Baseline

Combining algorithm:

Baseline

Combining algorithm:

- **BoW Q + I:** concatenate the BoW Q and I embeddings.

Baseline

Combining algorithm:

- **BoW Q + I:** concatenate the BoW Q and I embeddings.
- **LSTM Q + I:** Image embedding is first transformed to 1024-dim by a fully-connected layer. Then embeddings are fused via element-wise multiplication.

Baseline

Combining algorithm:

- **BoW Q + I:** concatenate the BoW Q and I embeddings.
- **LSTM Q + I:** Image embedding is first transformed to 1024-dim by a fully-connected layer. Then embeddings are fused via element-wise multiplication.

Implemenatation specific details:

Baseline

Combining algorithm:

- **BoW Q + I:** concatenate the BoW Q and I embeddings.
- **LSTM Q + I:** Image embedding is first transformed to 1024-dim by a fully-connected layer. Then embeddings are fused via element-wise multiplication.

Implemenatation specific details:

- The top K= 1000 most frequent answers are taken as possible outputs.

Baseline

Combining algorithm:

- **BoW Q + I:** concatenate the BoW Q and I embeddings.
- **LSTM Q + I:** Image embedding is first transformed to 1024-dim by a fully-connected layer. Then embeddings are fused via element-wise multiplication.

Implemenatation specific details:

- The top K= 1000 most frequent answers are taken as possible outputs.
- Combined image + question embedding is then passed to an 3 layer MLP with 1000 hidden units (dropout 0.5) in each layer and tanh non-linearity. Output layer is a softmax layer to obtain a distribution over K answers.

Baseline

Combining algorithm:

- **BoW Q + I:** concatenate the BoW Q and I embeddings.
- **LSTM Q + I:** Image embedding is first transformed to 1024-dim by a fully-connected layer. Then embeddings are fused via element-wise multiplication.

Implemenatation specific details:

- The top K= 1000 most frequent answers are taken as possible outputs.
- Combined image + question embedding is then passed to an 3 layer MLP with 1000 hidden units (dropout 0.5) in each layer and tanh non-linearity. Output layer is a softmax layer to obtain a distribution over K answers.
- VGGNet parameters are frozen.

Baseline

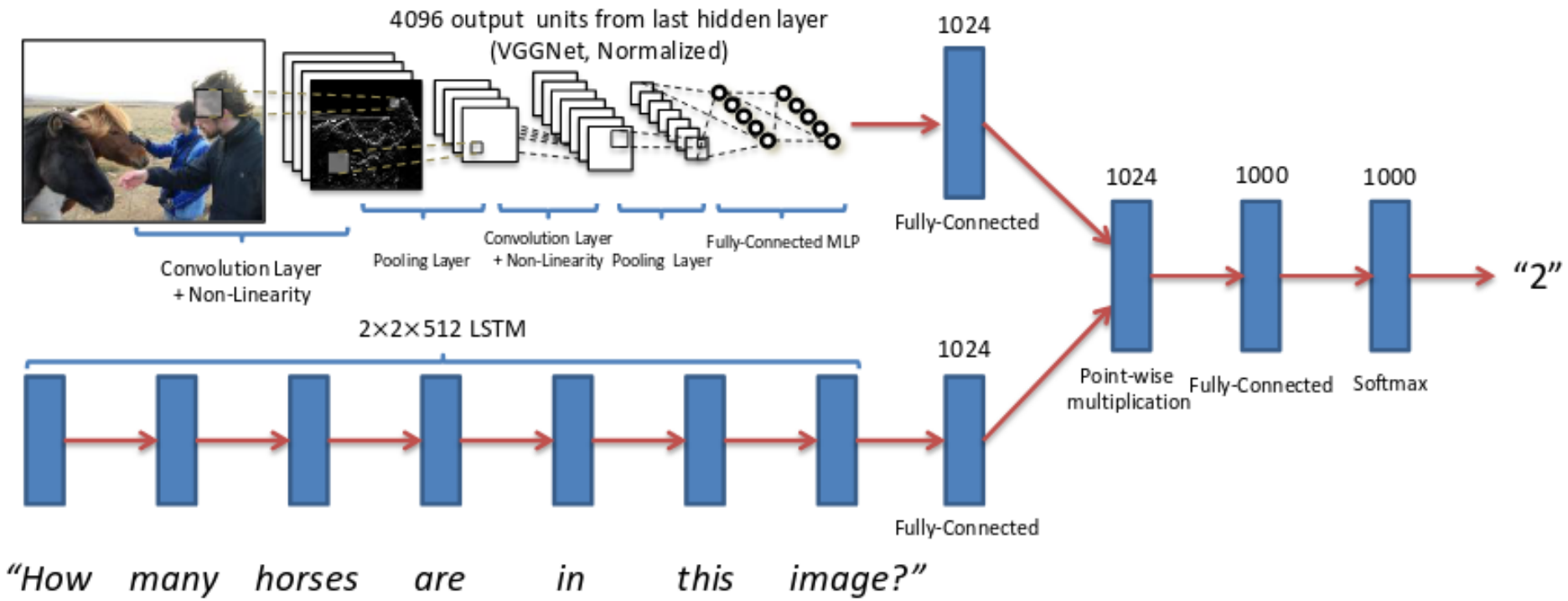
Combining algorithm:

- **BoW Q + I:** concatenate the BoW Q and I embeddings.
- **LSTM Q + I:** Image embedding is first transformed to 1024-dim by a fully-connected layer. Then embeddings are fused via element-wise multiplication.

Implemenatation specific details:

- The top K= 1000 most frequent answers are taken as possible outputs.
- Combined image + question embedding is then passed to an 3 layer MLP with 1000 hidden units (dropout 0.5) in each layer and tanh non-linearity. Output layer is a softmax layer to obtain a distribution over K answers.
- VGGNet parameters are frozen.
- Loss: End-to-end training with cross-entropy loss.

Baseline


4096 output units from last hidden layer
(VGGNet, Normalized)
1024
Fully-Connected
1024
1000
1000
Convolution Layer
+ Non-Linearity
Pooling Layer
Convolution Layer
+ Non-Linearity
Pooling Layer
Fully-Connected MLP
"2"
2×2×512 LSTM
1024
Point-wise
multiplication
Fully-Connected
Softmax
Fully-Connected
"How many horses are in this image?"

Baseline

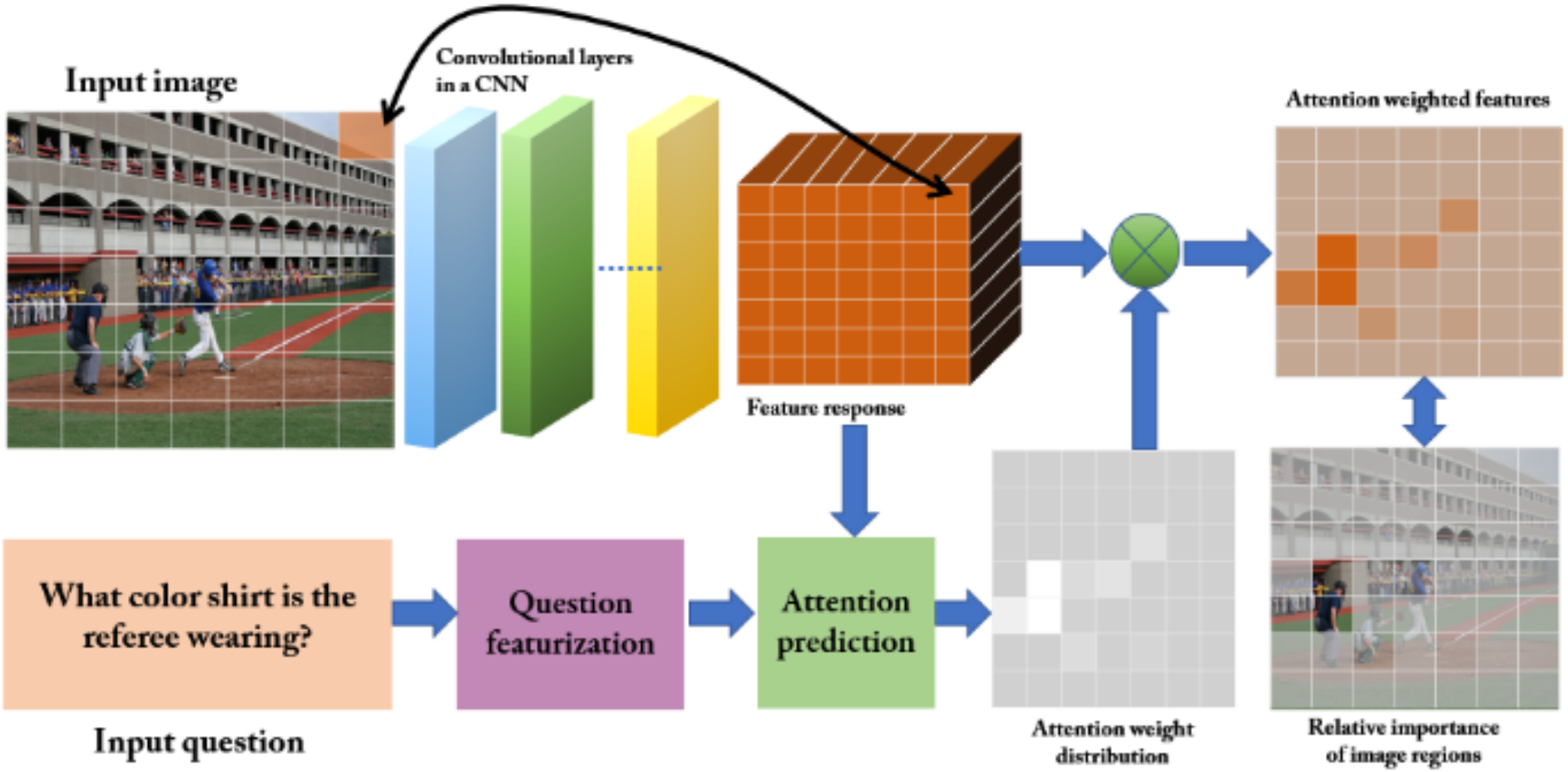
Table: Results

Model	**VQA**	
	OE	MC
Baseline	54.1	57.2

Spatial Attention based

Image features: VGGNet
Question features: LSTM
Combining algorithm: Use the question features to compute spatial attention maps for the visual features.

Spatial Attention based



Spatial Attention based

Spatial attention:

- $C(I)$ - feature map of image obtained from CNN.

Spatial Attention based

Spatial attention:

- $C(I)$ - feature map of image obtained from CNN.
- h_{t-1} - LSTM previous state

Spatial Attention based

Spatial attention:

- $C(I)$ - feature map of image obtained from CNN.
- h_{t-1} - LSTM previous state
- $e_t = w_a^T tanh(W_{he} h_{t-1} + W_{ce} C(I)) + b_a$

Spatial Attention based

Spatial attention:

- $C(I)$ - feature map of image obtained from CNN.
- h_{t-1} - LSTM previous state
- $e_t = w_a^T tanh(W_{he} h_{t-1} + W_{ce} C(I)) + b_a$
- $a_t = softmax(e_t)$

Spatial Attention based

Spatial attention:

- $C(I)$ - feature map of image obtained from CNN.
- h_{t-1} - LSTM previous state
- $e_t = w_a^T tanh(W_{he} h_{t-1} + W_{ce} C(I)) + b_a$
- $a_t = softmax(e_t)$
- $z_t = a_t^T C(I)$

Spatial Attention based

Spatial attention:

- $C(I)$ - feature map of image obtained from CNN.
- h_{t-1} - LSTM previous state
- $e_t = w_a^T tanh(W_{he} h_{t-1} + W_{ce} C(I)) + b_a$
- $a_t = softmax(e_t)$
- $z_t = a_t^T C(I)$

Stacked attention:

Spatial Attention based

Spatial attention:

- $C(I)$ - feature map of image obtained from CNN.
- h_{t-1} - LSTM previous state
- $e_t = w_a^T tanh(W_{he} h_{t-1} + W_{ce} C(I)) + b_a$
- $a_t = softmax(e_t)$
- $z_t = a_t^T C(I)$

Stacked attention:

- $z_t^1 = Att(h_{t-1}, C(I))$

Spatial Attention based

Spatial attention:

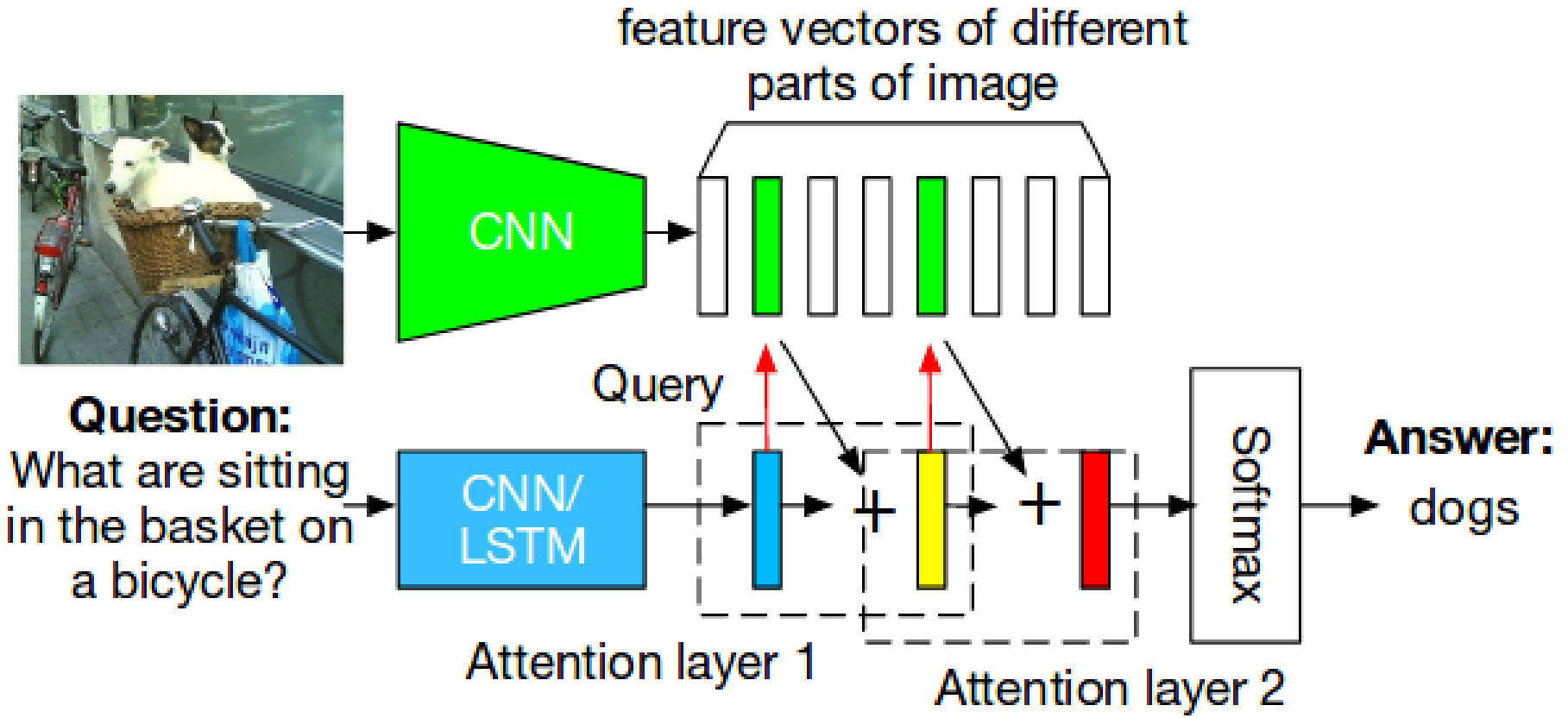
- $C(I)$ - feature map of image obtained from CNN.
- h_{t-1} - LSTM previous state
- $e_t = w_a^T tanh(W_{he} h_{t-1} + W_{ce} C(I)) + b_a$
- $a_t = softmax(e_t)$
- $z_t = a_t^T C(I)$

Stacked attention:

- $z_t^1 = Att(h_{t-1}, C(I))$
- $z_t^2 = Att(h_{t-1}, C(I), z_t^1)$

Spatial Attention based

Architecture:



Spatial Attention based

Example:

(a) What are pulling a man on a wagon down on dirt road?
Answer: horses Prediction: horses

(c) What next to the large umbrella attached to a table?
Answer: trees Prediction: tree

Spatial Attention based

- Training: stochastic gradient descent with momentum 0.9.
- Batch size: 100
- Gradient clipping and dropout used.
- LSTM dimension: 1000, CNN-Unigram filter: 128, CNN-Bigram filter: 256, CNN-Trigram filter: 256, CNN dimension: 640

Spatial attention based

Table: Results

Model	**VQA**	
	OE	MC
Baseline	54.1	57.2
Stacked attention	58.9	-

Joint Attention based

Image features: VGGNet/ResNet
Question features: Bidirectional LSTM
Combining algorithm: Apply attention to both the image and question to jointly reason about the two different streams of information.

Joint Attention based

- Allow image and question attention to guide each other, directing attention to relevant words and visual regions simultaneously.

Joint Attention based

- Allow image and question attention to guide each other, directing attention to relevant words and visual regions simultaneously.
- Visual and question input are jointly represented by a memory vector that is used to simultaneously predict attention for both question and image features.

Joint Attention based

- Allow image and question attention to guide each other, directing attention to relevant words and visual regions simultaneously.
- Visual and question input are jointly represented by a memory vector that is used to simultaneously predict attention for both question and image features.
- Use updated representations to recursively update the memory vector.

Joint Attention based

Image representation:

- The image features are extracted from 19-layer VGGNet or 152-layer ResNet.

Joint Attention based

Image representation:

- The image features are extracted from 19-layer VGGNet or 152-layer ResNet.
- The image is represented by $\{v_1, \ldots, v_N\}$, where N is the number of image regions.

Joint Attention based

Image representation:

- The image features are extracted from 19-layer VGGNet or 152-layer ResNet.
- The image is represented by $\{v_1, \ldots, v_N\}$, where N is the number of image regions.

Question features:

Joint Attention based

Image representation:

- The image features are extracted from 19-layer VGGNet or 152-layer ResNet.
- The image is represented by $\{v_1, \ldots, v_N\}$, where N is the number of image regions.

Question features:

- Bidirectional LSTMs

$$h_t^{(f)} = LSTM^{(f)}(x_t, h_{t-1}^{(f)})$$
$$h_t^{(b)} = LSTM^{(b)}(x_t, h_{t-1}^{(b)})$$
$$u_t = h_t^{(f)} + h_t^{(b)}$$

Joint Attention based

Image representation:

- The image features are extracted from 19-layer VGGNet or 152-layer ResNet.
- The image is represented by $\{v_1, \ldots, v_N\}$, where N is the number of image regions.

Question features:

- Bidirectional LSTMs

$$h_t^{(f)} = LSTM^{(f)}(x_t, h_{t-1}^{(f)})$$
$$h_t^{(b)} = LSTM^{(b)}(x_t, h_{t-1}^{(b)})$$
$$u_t = h_t^{(f)} + h_t^{(b)}$$

- $\{u_1, \ldots, u_T\}$ is the representation of the sentence.

Joint Attention based

Notation:

- v^k - attended image representation at k-th step.

Joint Attention based

Notation:

- v^k - attended image representation at k-th step.
- u^k - attended question representation at k-th step.

Joint Attention based

Notation:

- v^k - attended image representation at k-th step.
- u^k - attended question representation at k-th step.
- m^k - attended combined representation at k-th step.

Joint Attention based

Notation:

- v^k - attended image representation at k-th step.
- u^k - attended question representation at k-th step.
- m^k - attended combined representation at k-th step.
- P^k - weight matrix to transform from image space to question space.

Joint Attention based

Notation:

- v^k - attended image representation at k-th step.
- u^k - attended question representation at k-th step.
- m^k - attended combined representation at k-th step.
- P^k - weight matrix to transform from image space to question space.

Method:

Joint Attention based

Notation:

- v^k - attended image representation at k-th step.
- u^k - attended question representation at k-th step.
- m^k - attended combined representation at k-th step.
- P^k - weight matrix to transform from image space to question space.

Method:

- Start with equal attention at all - $u^{(0)} = \frac{1}{T} \sum_t u_t$

Joint Attention based

Notation:

- v^k - attended image representation at k-th step.
- u^k - attended question representation at k-th step.
- m^k - attended combined representation at k-th step.
- P^k - weight matrix to transform from image space to question space.

Method:

- Start with equal attention at all - $u^{(0)} = \frac{1}{T} \sum_t u_t$
- $v^0 = tanh(P^0 \frac{1}{N} \sum_n u_n)$

Joint Attention based

Notation:

- v^k - attended image representation at k-th step.
- u^k - attended question representation at k-th step.
- m^k - attended combined representation at k-th step.
- P^k - weight matrix to transform from image space to question space.

Method:

- Start with equal attention at all - $u^{(0)} = \frac{1}{T} \sum_t u_t$
- $v^0 = tanh(P^0 \frac{1}{N} \sum_n u_n)$
- $m^0 = v^0 \odot u^0$

Joint Attention based

Notation:

- v^k - attended image representation at k-th step.
- u^k - attended question representation at k-th step.
- m^k - attended combined representation at k-th step.
- P^k - weight matrix to transform from image space to question space.

Method:

- Start with equal attention at all - $u^{(0)} = \frac{1}{T} \sum_t u_t$
- $v^0 = tanh(P^0 \frac{1}{N} \sum_n u_n)$
- $m^0 = v^0 \odot u^0$
- $m^k = m^{k-1} + v^k \odot u^k$

Joint Attention based

Visual attention:

Joint Attention based

Visual attention:

- Attention to get representation at k-th step
  $v^k = VAtt(\{v_n\}_{n=1}^N, m^{k-1})$

Joint Attention based

Visual attention:

- Attention to get representation at k-th step
  $v^k = VAtt(\{v_n\}_{n=1}^N, m^{k-1})$
- VAtt is 2 layer network.

Joint Attention based

Visual attention:

- Attention to get representation at k-th step
  $v^k = VAtt(\{v_n\}_{n=1}^N, m^{k-1})$
- VAtt is 2 layer network.
- $h_{vn}^k = tanh(W_v^k v_n) \odot tanh(W_{vm}^k m^{k-1})$

Joint Attention based

Visual attention:

- Attention to get representation at k-th step
  $v^k = VAtt(\{v_n\}_{n=1}^N, m^{k-1})$
- VAtt is 2 layer network.
- $h_{vn}^k = tanh(W_v^k v_n) \odot tanh(W_{vm}^k m^{k-1})$
- $\alpha_{vn}^k = softmax(W_{vh}^k h_{vn}^k)$

Joint Attention based

Visual attention:

- Attention to get representation at k-th step
  $v^k = VAtt(\{v_n\}_{n=1}^N, m^{k-1})$
- VAtt is 2 layer network.
- $h_{vn}^k = tanh(W_v^k v_n) \odot tanh(W_{vm}^k m^{k-1})$
- $\alpha_{vn}^k = softmax(W_{vh}^k h_{vn}^k)$
- $v^k = tanh(P^k \sum_{n=1}^N \alpha_{vn}^k v_n)$

Joint Attention based

Visual attention:

- Attention to get representation at k-th step
  $v^k = VAtt(\{v_n\}_{n=1}^N, m^{k-1})$
- VAtt is 2 layer network.
- $h_{vn}^k = tanh(W_v^k v_n) \odot tanh(W_{vm}^k m^{k-1})$
- $\alpha_{vn}^k = softmax(W_{vh}^k h_{vn}^k)$
- $v^k = tanh(P^k \sum_{n=1}^N \alpha_{vn}^k v_n)$

Textual attention:

Joint Attention based

Visual attention:

- Attention to get representation at k-th step
  $v^k = VAtt(\{v_n\}_{n=1}^N, m^{k-1})$
- VAtt is 2 layer network.
- $h_{vn}^k = tanh(W_v^k v_n) \odot tanh(W_{vm}^k m^{k-1})$
- $\alpha_{vn}^k = softmax(W_{vh}^k h_{vn}^k)$
- $v^k = tanh(P^k \sum_{n=1}^N \alpha_{vn}^k v_n)$

Textual attention:

- Textual attention is also similar.
  $u^k = TAtt(\{u_t\}_{t=1}^T, m^{k-1})$

Joint Attention based

Visual attention:

- Attention to get representation at k-th step
  $v^k = VAtt(\{v_n\}_{n=1}^N, m^{k-1})$
- VAtt is 2 layer network.
- $h_{vn}^k = tanh(W_v^k v_n) \odot tanh(W_{vm}^k m^{k-1})$
- $\alpha_{vn}^k = softmax(W_{vh}^k h_{vn}^k)$
- $v^k = tanh(P^k \sum_{n=1}^N \alpha_{vn}^k v_n)$

Textual attention:

- Textual attention is also similar.
  $u^k = TAtt(\{u_t\}_{t=1}^T, m^{k-1})$
- $h_{ut}^k = tanh(W_u^k u_t) \odot tanh(W_{um}^k m^{k-1})$

Joint Attention based

Visual attention:

- Attention to get representation at k-th step
  $v^k = VAtt(\{v_n\}_{n=1}^N, m^{k-1})$
- VAtt is 2 layer network.
- $h_{vn}^k = tanh(W_v^k v_n) \odot tanh(W_{vm}^k m^{k-1})$
- $\alpha_{vn}^k = softmax(W_{vh}^k h_{vn}^k)$
- $v^k = tanh(P^k \sum_{n=1}^N \alpha_{vn}^k v_n)$

Textual attention:

- Textual attention is also similar.
  $u^k = TAtt(\{u_t\}_{t=1}^T, m^{k-1})$
- $h_{ut}^k = tanh(W_u^k u_t) \odot tanh(W_{um}^k m^{k-1})$
- $\alpha_{ut}^k = softmax(W_{uh}^k h_{ut}^k)$

Joint Attention based

Visual attention:

- Attention to get representation at k-th step
  $v^k = VAtt(\{v_n\}_{n=1}^N, m^{k-1})$
- VAtt is 2 layer network.
- $h_{vn}^k = tanh(W_v^k v_n) \odot tanh(W_{vm}^k m^{k-1})$
- $\alpha_{vn}^k = softmax(W_{vh}^k h_{vn}^k)$
- $v^k = tanh(P^k \sum_{n=1}^N \alpha_{vn}^k v_n)$

Textual attention:

- Textual attention is also similar.
  $u^k = TAtt(\{u_t\}_{t=1}^T, m^{k-1})$
- $h_{ut}^k = tanh(W_u^k u_t) \odot tanh(W_{um}^k m^{k-1})$
- $\alpha_{ut}^k = softmax(W_{uh}^k h_{ut}^k)$
- $u^k = \sum_t \alpha_{ut}^k u_t$

Joint Attention based

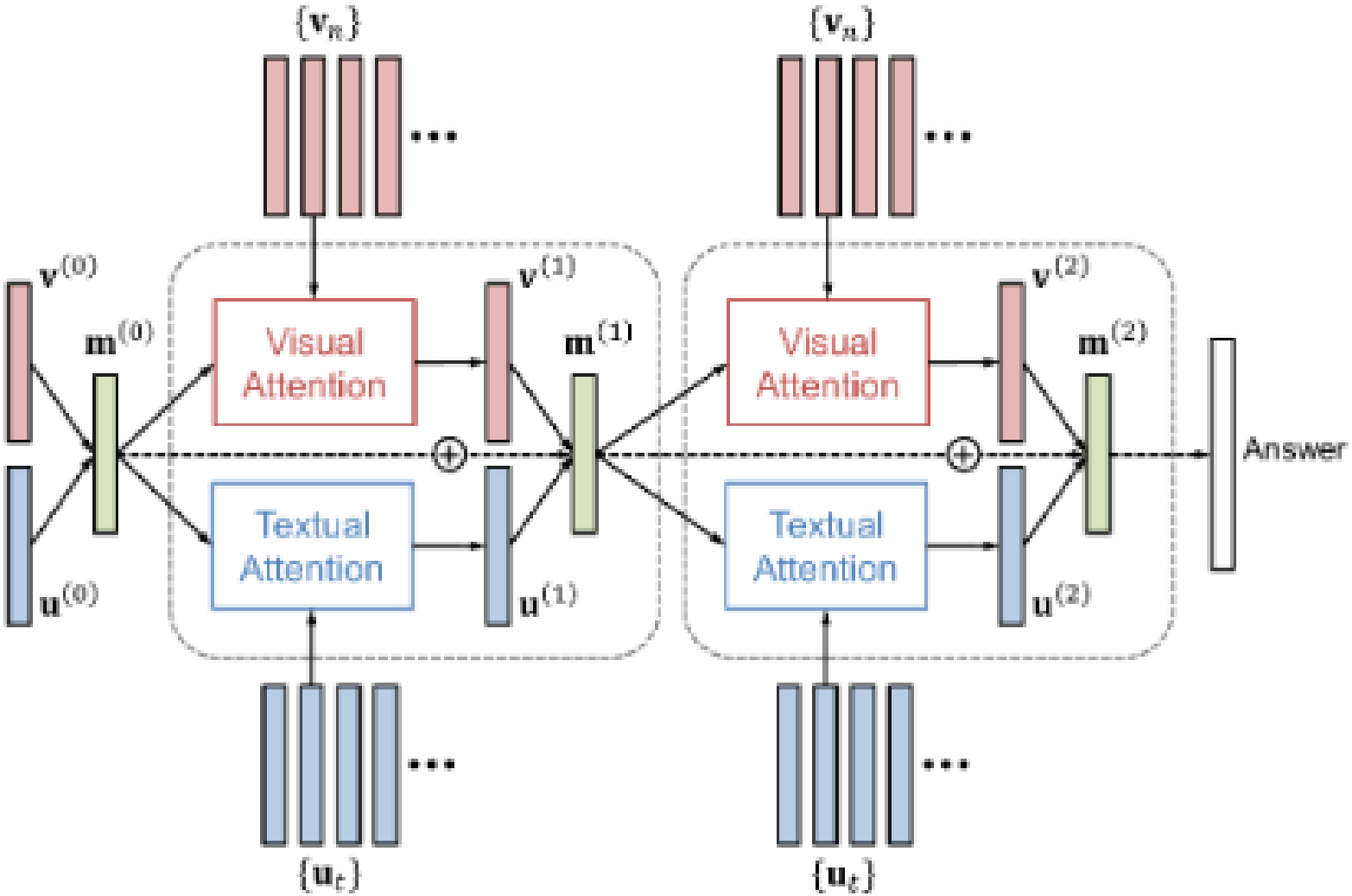


Figure: VQA architecture

Joint Attention based

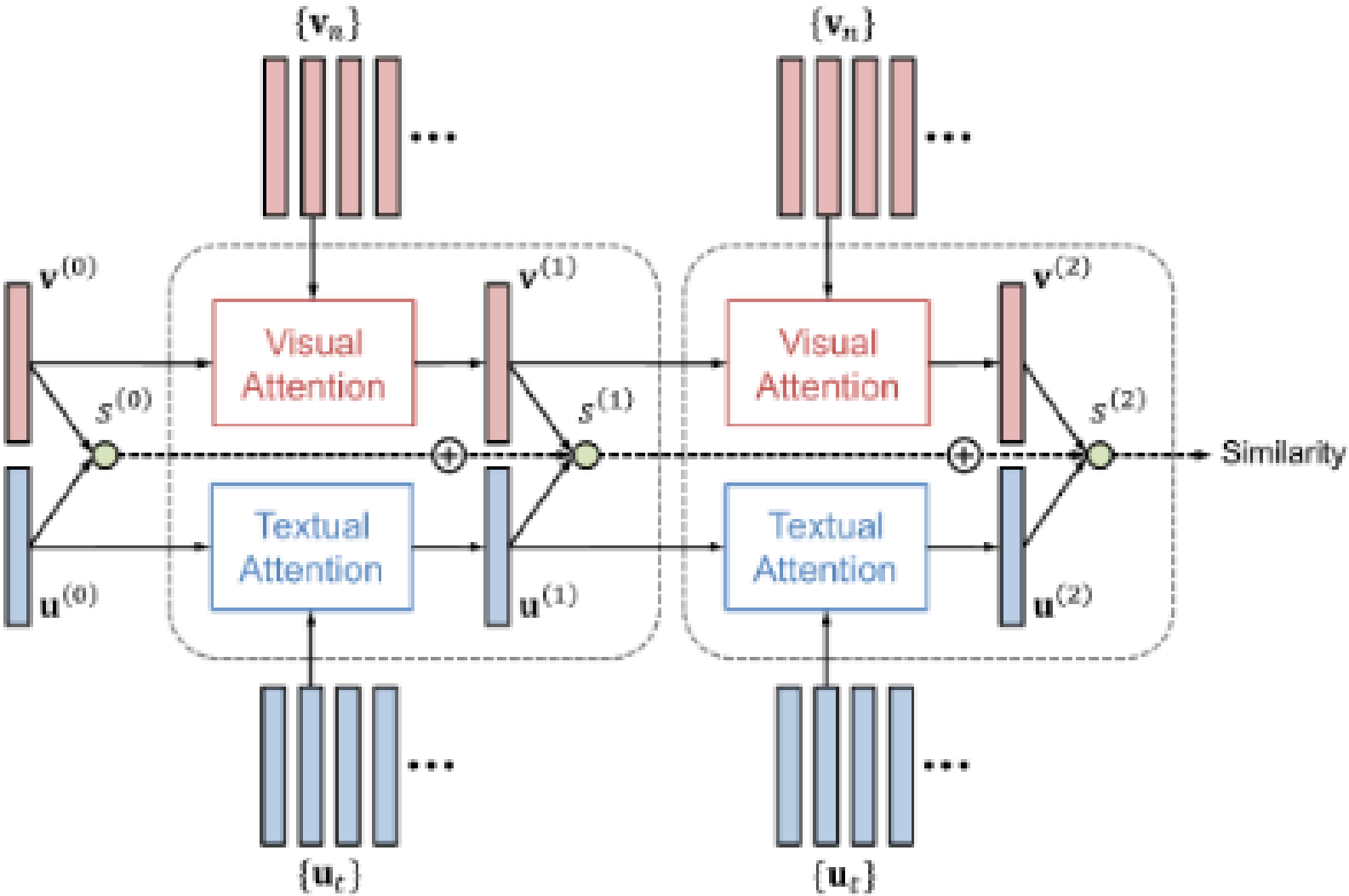


Figure: Image-text matching architecture

Joint Attention based

Implementation details:

- $K = 2$
- Text embedding size: 512
- Training: SGD with a learning rate 0.1, momentum 0.9, weight decay 0.0005, dropout ratio 0.5, and gradient clipping at 0.1 for 60 epochs.
- Batch size: 128
- Top 2000 answers are considered as candidates.

Joint Attention based

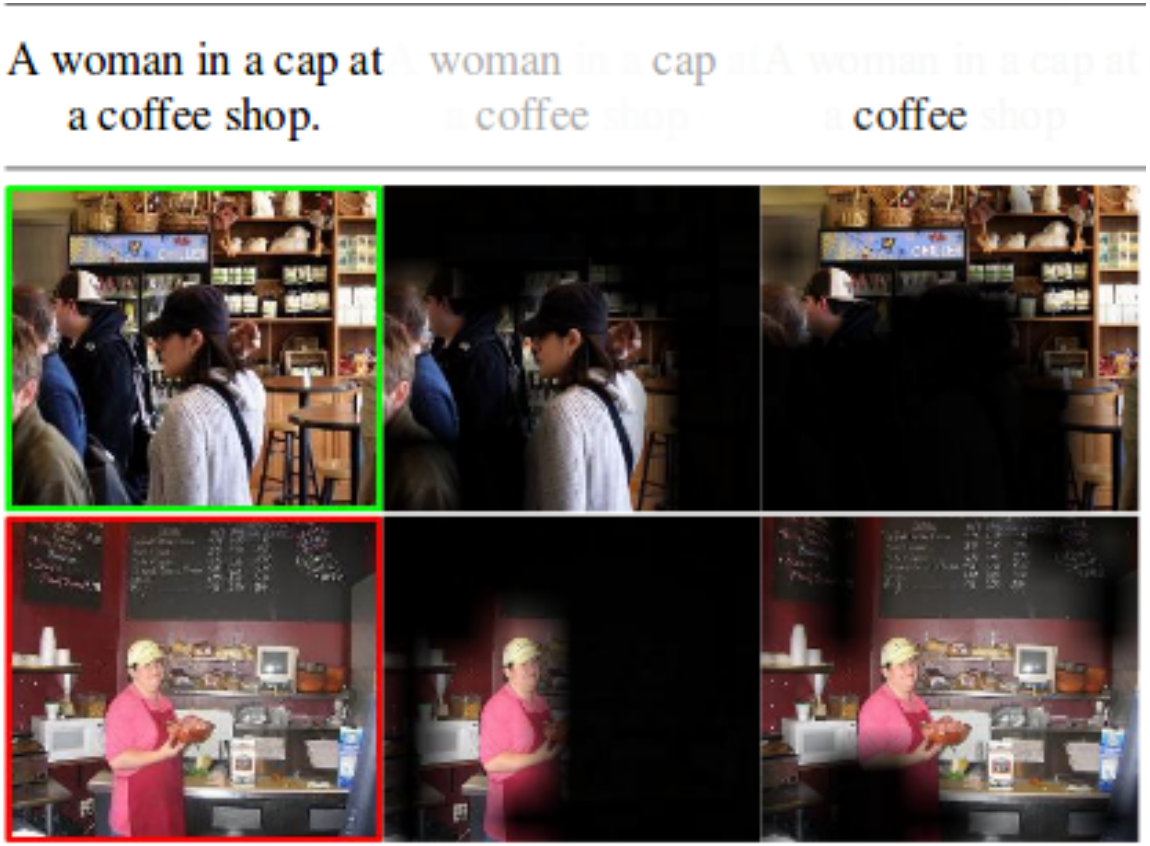
Example:

Q: How many horses are in the picture?
A: 2

Q: What is on his wrist?
A: watch

Joint Attention based

Text-to-image retrieval:



Joint Attention based

Image-to-text retrieval:

(+) Two boys playing together at a playground.	boys playground	playground
(-) The two kids are playing at the playground.	kids playground	playground

Joint Attention based

Table: Results

Model	**VQA**	
	OE	MC
Baseline	54.1	57.2
Stacked attention	58.9	-
Joint attention	64.2	69.0

Bilinear pooling based

Image features: ResNet
Question features: GRU
Combining algorithm: Instead of simple functions use more complex methods to combine image and question features. Outer-product is one example of complex interaction.

Bilinear pooling based

Bilinear pooling:

- The key problem in VQA is combining the information obtained from the 2 feature spaces.

Bilinear pooling based

Bilinear pooling:

- The key problem in VQA is combining the information obtained from the 2 feature spaces.
- Outer-product of two vectors in different spaces will capture all the complex interactions between them. But it is computationally expensive to compute.

Bilinear pooling based

Bilinear pooling:

- The key problem in VQA is combining the information obtained from the 2 feature spaces.
- Outer-product of two vectors in different spaces will capture all the complex interactions between them. But it is computationally expensive to compute.
- So, the idea is to approximate the outer-product between the image and text features, allowing a deeper interaction between the two modalities.

Bilinear pooling based

Bilinear pooling:

- The key problem in VQA is combining the information obtained from the 2 feature spaces.
- Outer-product of two vectors in different spaces will capture all the complex interactions between them. But it is computationally expensive to compute.
- So, the idea is to approximate the outer-product between the image and text features, allowing a deeper interaction between the two modalities.
- Rather than doing the outer-product explicitly, which would be very high dimensional, its done in a lower dimensional space.

Bilinear pooling based

Bilinear pooling:

- The key problem in VQA is combining the information obtained from the 2 feature spaces.
- Outer-product of two vectors in different spaces will capture all the complex interactions between them. But it is computationally expensive to compute.
- So, the idea is to approximate the outer-product between the image and text features, allowing a deeper interaction between the two modalities.
- Rather than doing the outer-product explicitly, which would be very high dimensional, its done in a lower dimensional space.
- The bilinear form allows the outputs of the feature extractors to be conditioned on each other by considering all their pairwise interactions similar to a quadratic kernel expansion.

Bilinear pooling based

Dimension problem:

Bilinear pooling based

Dimension problem:

- y: predicted answer
- q: question feature vector
- v: Image feature vector
- W: weight tensor
- $y = (W \times q) \times v$, $v \in R^N$, $q \in R^T$, $y \in R^K$, $W \in R^{T \times N \times K}$

Bilinear pooling based

Dimension problem:

- y: predicted answer
- q: question feature vector
- v: Image feature vector
- W: weight tensor
- $y = (W \times q) \times v$, $v \in R^N$, $q \in R^T$, $y \in R^K$, $W \in R^{T \times N \times K}$
- $T, N, y \sim 1000$, No. of parameters: 10^9

Bilinear pooling based

Dimension problem:

- y: predicted answer
- q: question feature vector
- v: Image feature vector
- W: weight tensor
- $y = (W \times q) \times v$, $v \in R^N$, $q \in R^T$, $y \in R^K$, $W \in R^{T \times N \times K}$
- $T, N, y \sim 1000$, No. of parameters: 10^9

Solution:

Bilinear pooling based

Dimension problem:

- y: predicted answer
- q: question feature vector
- v: Image feature vector
- W: weight tensor
- $y = (W \times q) \times v$, $v \in R^N$, $q \in R^T$, $y \in R^K$, $W \in R^{T \times N \times K}$
- $T, N, y \sim 1000$, No. of parameters: 10^9

Solution:

- Factorise

Bilinear pooling based

Dimension problem:

- y: predicted answer
- q: question feature vector
- v: Image feature vector
- W: weight tensor
- $y = (W \times q) \times v$, $v \in R^N$, $q \in R^T$, $y \in R^K$, $W \in R^{T \times N \times K}$
- $T, N, y \sim 1000$, No. of parameters: 10^9

Solution:

- Factorise
- $W = ((W_c \times W_q) \times W_v) \times W_o$

Bilinear pooling based

Dimension problem:

- y: predicted answer
- q: question feature vector
- v: Image feature vector
- W: weight tensor
- $y = (W \times q) \times v$, $v \in R^N$, $q \in R^T$, $y \in R^K$, $W \in R^{T \times N \times K}$
- $T, N, y \sim 1000$, No. of parameters: 10^9

Solution:

- Factorise
- $W = ((W_c \times W_q) \times W_v) \times W_o$
- $W_q \in R^{T \times t}$, $W_v \in R^{N \times n}$, $W_o \in R^{K \times k}$, $W_c \in R^{t \times n \times k}$.

Bilinear pooling based

Dimension problem:

- y: predicted answer
- q: question feature vector
- v: Image feature vector
- W: weight tensor
- $y = (W \times q) \times v$, $v \in R^N$, $q \in R^T$, $y \in R^K$, $W \in R^{T \times N \times K}$
- $T, N, y \sim 1000$, No. of parameters: 10^9

Solution:

- Factorise
- $W = ((W_c \times W_q) \times W_v) \times W_o$
- $W_q \in R^{T \times t}$, $W_v \in R^{N \times n}$, $W_o \in R^{K \times k}$, $W_c \in R^{t \times n \times k}$.
- $y = ((W_c \times (q^T W_q)) \times (v^T W_v)) \times W_o$

Bilinear pooling based

Add sparsity:

- $\widetilde{q} = q^T W_q$

Bilinear pooling based

Add sparsity:

- $\widetilde{q} = q^T W_q$
- $\widetilde{v} = v^T W_v$

Bilinear pooling based

Add sparsity:

- $\widetilde{q} = q^T W_q$
- $\widetilde{v} = v^T W_v$
- Consider $z[p] = \widetilde{q}^T W_c[:, :, p]\widetilde{v}$

Bilinear pooling based

Add sparsity:

- $\widetilde{q} = q^T W_q$
- $\widetilde{v} = v^T W_v$
- Consider $z[p] = \widetilde{q}^T W_c[:, :, p]\widetilde{v}$
- Impose rank of each slice of tensor must be R

Bilinear pooling based

Add sparsity:

- $\widetilde{q} = q^T W_q$
- $\widetilde{v} = v^T W_v$
- Consider $z[p] = \widetilde{q}^T W_c[:, :, p]\widetilde{v}$
- Impose rank of each slice of tensor must be R
- $W_c[:, :, p] = \sum_{r=1}^{R} m_r^p \otimes n_r^p$

Bilinear pooling based

Add sparsity:

- $\widetilde{q} = q^T W_q$
- $\widetilde{v} = v^T W_v$
- Consider $z[p] = \widetilde{q}^T W_c[:, :, p]\widetilde{v}$
- Impose rank of each slice of tensor must be R
- $W_c[:, :, p] = \sum_{r=1}^{R} m_r^p \otimes n_r^p$
- $z[p] = \sum_{r=1}^{R} (\widetilde{q}^T m_r^p)(\widetilde{v}^T n_r^p)$

Bilinear pooling based

Add sparsity:

- $\widetilde{q} = q^T W_q$
- $\widetilde{v} = v^T W_v$
- Consider $z[p] = \widetilde{q}^T W_c[:, :, p]\widetilde{v}$
- Impose rank of each slice of tensor must be R
- $W_c[:, :, p] = \sum_{r=1}^{R} m_r^p \otimes n_r^p$
- $z[p] = \sum_{r=1}^{R} (\widetilde{q}^T m_r^p)(\widetilde{v}^T n_r^p)$
- $z = \sum_{r=1}^{R} z_r$

Bilinear pooling based

Add sparsity:

- $\widetilde{q} = q^T W_q$
- $\widetilde{v} = v^T W_v$
- Consider $z[p] = \widetilde{q}^T W_c[:, :, p]\widetilde{v}$
- Impose rank of each slice of tensor must be R
- $W_c[:, :, p] = \sum_{r=1}^{R} m_r^p \otimes n_r^p$
- $z[p] = \sum_{r=1}^{R} (\widetilde{q}^T m_r^p)(\widetilde{v}^T n_r^p)$
- $z = \sum_{r=1}^{R} z_r$
- $z_r = (\widetilde{q}^T M_r)(\widetilde{v}^T N_r)$

Bilinear pooling based

Add sparsity:

- $\widetilde{q} = q^T W_q$
- $\widetilde{v} = v^T W_v$
- Consider $z[p] = \widetilde{q}^T W_c[:, :, p]\widetilde{v}$
- Impose rank of each slice of tensor must be R
- $W_c[:, :, p] = \sum_{r=1}^{R} m_r^p \otimes n_r^p$
- $z[p] = \sum_{r=1}^{R} (\widetilde{q}^T m_r^p)(\widetilde{v}^T n_r^p)$
- $z = \sum_{r=1}^{R} z_r$
- $z_r = (\widetilde{q}^T M_r)(\widetilde{v}^T N_r)$
- $M_r \in R^{t \times k}$, $N_r \in R^{n \times k}$

Bilinear pooling based

Add sparsity:

- $\widetilde{q} = q^T W_q$
- $\widetilde{v} = v^T W_v$
- Consider $z[p] = \widetilde{q}^T W_c[:, :, p]\widetilde{v}$
- Impose rank of each slice of tensor must be R
- $W_c[:, :, p] = \sum_{r=1}^{R} m_r^p \otimes n_r^p$
- $z[p] = \sum_{r=1}^{R} (\widetilde{q}^T m_r^p)(\widetilde{v}^T n_r^p)$
- $z = \sum_{r=1}^{R} z_r$
- $z_r = (\widetilde{q}^T M_r)(\widetilde{v}^T N_r)$
- $M_r \in R^{t \times k}$, $N_r \in R^{n \times k}$
- Final parameters: W_o, W_q, W_v, $\{M_r\}_{r=1}^{R}$, $\{N_r\}_{r=1}^{R}$

Bilinear pooling based

Add sparsity:

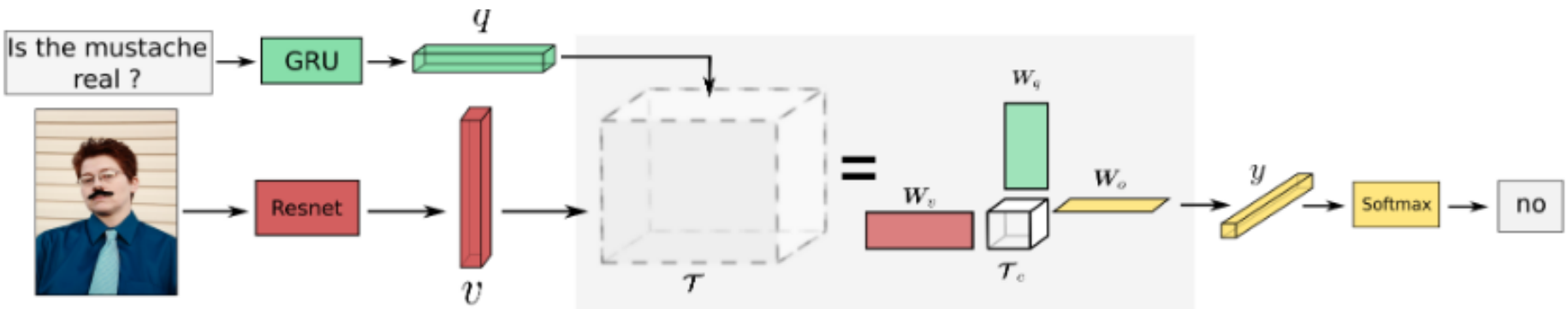
- $\widetilde{q} = q^T W_q$
- $\widetilde{v} = v^T W_v$
- Consider $z[p] = \widetilde{q}^T W_c[:, :, p]\widetilde{v}$
- Impose rank of each slice of tensor must be R
- $W_c[:, :, p] = \sum_{r=1}^{R} m_r^p \otimes n_r^p$
- $z[p] = \sum_{r=1}^{R} (\widetilde{q}^T m_r^p)(\widetilde{v}^T n_r^p)$
- $z = \sum_{r=1}^{R} z_r$
- $z_r = (\widetilde{q}^T M_r)(\widetilde{v}^T N_r)$
- $M_r \in R^{t \times k}$, $N_r \in R^{n \times k}$
- Final parameters: W_o, W_q, W_v, $\{M_r\}_{r=1}^{R}$, $\{N_r\}_{r=1}^{R}$
- Interpretation: $z_r[p] = (\widetilde{q}$ similar to $m_r^p)$ AND $(\widetilde{v}$ similar to $n_r^p)$

Bilinear pooling based

Add sparsity:

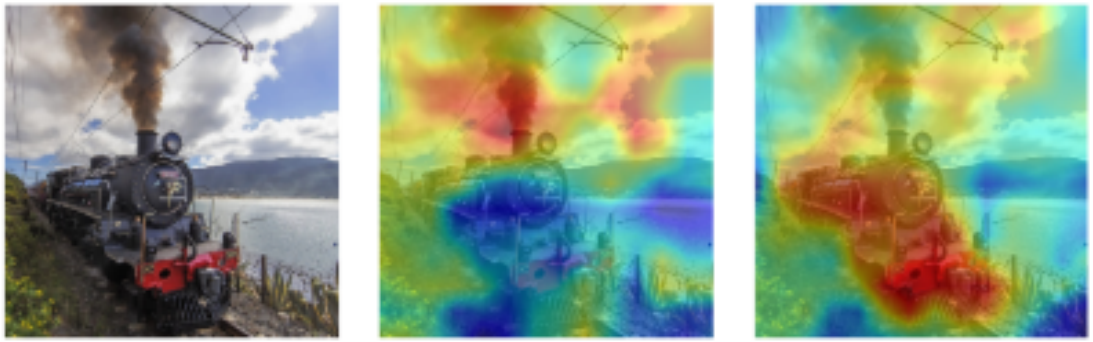
- $\widetilde{q} = q^T W_q$
- $\widetilde{v} = v^T W_v$
- Consider $z[p] = \widetilde{q}^T W_c[:, :, p]\widetilde{v}$
- Impose rank of each slice of tensor must be R
- $W_c[:, :, p] = \sum_{r=1}^{R} m_r^p \otimes n_r^p$
- $z[p] = \sum_{r=1}^{R} (\widetilde{q}^T m_r^p)(\widetilde{v}^T n_r^p)$
- $z = \sum_{r=1}^{R} z_r$
- $z_r = (\widetilde{q}^T M_r)(\widetilde{v}^T N_r)$
- $M_r \in R^{t \times k}$, $N_r \in R^{n \times k}$
- Final parameters: W_o, W_q, W_v, $\{M_r\}_{r=1}^{R}$, $\{N_r\}_{r=1}^{R}$
- Interpretation: $z_r[p] = (\widetilde{q}$ similar to $m_r^p)$ AND $(\widetilde{v}$ similar to $n_r^p)$
- $z[p] = z_1[p]$ OR $\ldots$ OR $z_R[p]$

Bilinear pooling based



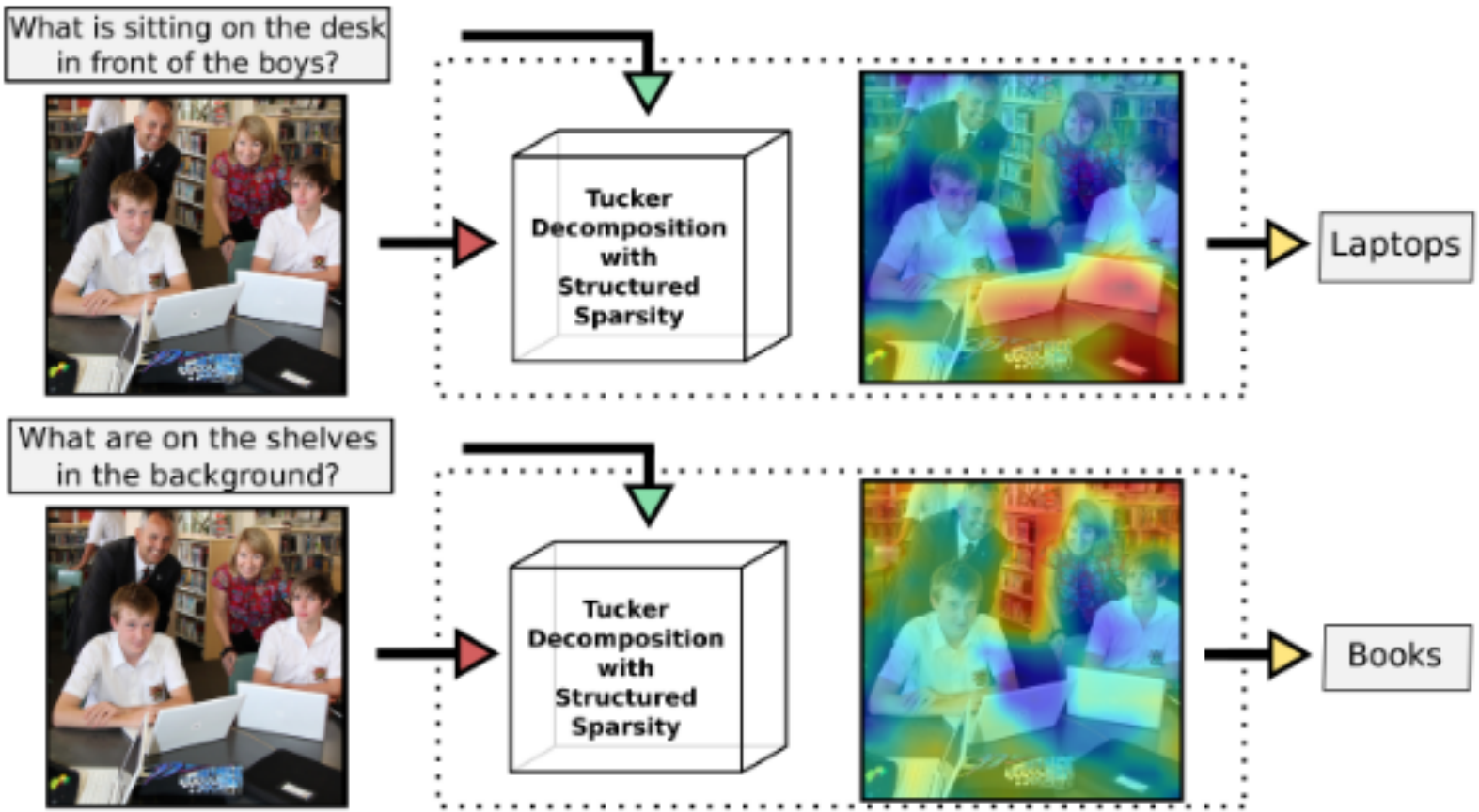
Bilinear pooling based

(a) Question: Where is the woman ? - Answer: on the elephant



(b) Question: Where is the smoke coming from ? - Answer: train

Bilinear pooling based



Bilinear pooling based

Implementation deatils:

- Attention can still be used to obtain v.
- t=n=k=360, R=10
- Top-2000 answers are considered as candidates.
- Training is done using Adam,
- Batch size - 512

Bilinear pooling based

Table: Results

Model	**VQA**	
	OE	MC
Baseline	54.1	57.2
Stacked attention	58.9	-
Joint attention	64.2	69.0
Bilinear	60.1	-
Binilear Ensemble	67.3	-

Thank You.