Generative Question Answering: Learning to Answer the Whole Question

M. Lewis, A. Fan

Facebook AI Research

Presenter: Bill Zhang

https://qdata.github.io/deep2Read

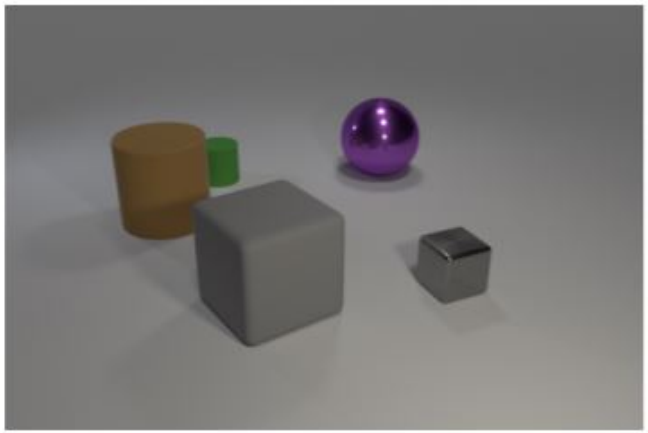
Outline

Introduction

- Current models for QA achieve high performance with superficial understanding
  - In Visual QA, "what color is the grass?" can be answered without image
- Over-fitting to biases caused by discriminitive loss functions, which saturate when simple correlations allow for confident answers
- Propose generative QA model, using Bayes' rule to get questions given answers
- Generative loss functions train model to explain all question words

Introduction

Problem Illustration



Is the purple thing the same shape as the large gray rubber thing?
Does the green rubber object have the same shape as the gray thing that is on the right side of the big purple object?

Whilst filming in Mexico City, speculation in the media claimed that the script had been altered to accommodate the demands of Mexican authorities reportedly influencing details of the scene and characters, casting choices, and modifying the script in order to portray the country in a "positive light" in order to secure tax concessions and financial support worth up to $20 million for the film. This was denied by producer Michael G. Wilson.

Which Bond producer would not confirm that the film had been changed to accommodate Mexican authorities?

Model

Overview

- Dataset of examples with questions $q = q_{0...T}$, answer a, and context c (image or text)
- Train models to optimize

$$\mathcal{L} = -\log p(q, a|c) = -\log p(a|c) - \Sigma_t \log p(q_t|a, c, q_{0...t-1})$$

- First, encode c using RNN (text) or CNN (image)
- Prior $p(a|c)$ evaluated by scoring all answers
- Likelihood $p(q|a, c)$ is modelled using conditional language model
- Answer maximizing $p(q, a|c)$ returned: $\text{argmax}_a p(q, a|c) = \text{argmax}_a p(a|q, c)$

Model

Document Encoder: Answer-independent Context Representations

- Character-based word embeddings and train 2-layer LSTM language model in forward and reverse directions using WikiText-103 corpus
- Parameters frozen and not fine-tuned
- Contextualized word representations computed as fully connected layer applied to concatenation of word embedding and hidden states of each layer
- Sum representation with trainable vector of size d if word occurs in article title
- Follow with residual bidirectional LSTMs of size $d/2$, concatenating output in each direction

Model

Document Encoder: Answer Encoder

- Assume answers are a span $i...j$ of the document
- Answer represented as weighted sum of words in span
- For each word representation $\sum_{k=i...j} \sigma(w\tilde{c}_k)\tilde{c}_k$, $w \in \mathbb{R}^d$ is a trainable vector and $\tilde{c}_k$ is the kth word in the answer-independent document representation
- This allows selection of multiple head words in a span

Model

Document Encoder: Answer-dependent Context Representation

- Model more complex interactions between answer and context because only correct answer is used
- Take answer-independent representation of each context word
- Concatenate with 32-dimensional embeddings of
  - Binary feature for whether or not word is in answer
  - Position relative to answer start
  - Position relative to answer end
  - Element-wise product of word representation and answer encoding
- Feed into 3 further layers of residual bidirectional LSTMs of size $d/2$

Model

Image Encoder

- Simple image encoder, leaving reasoning to question decoder
- Like Johnson et al. (2017), take pre-trained features from conv4 layer ResNet-101 model
- 14x14 grid, 1024-dimensional features, dropout, project representations to size d with 1x1 convolution, batch normalization, ReLU activation,...
- Concatenate final representation with 32-dimensional positional encoding

Model

Answer Prior: SQuAD

- On SQuAD, concatenate the start and end representations of answer-independent context representation, combine them with single hidden layer of size $2d$ and ReLU activation, and project down to score $s^{\text{endpoints}}(a, c)$
- Then, add additional score based on answer length $s^{\text{length}}(a)$
- $p(a|c) = \frac{\exp(s^{\text{endpoints}}(a,c)+s^{\text{length}})(a)}{\Sigma_{a'} \exp(s^{\text{endpoints}}(a',c)+s^{\text{length}}(a'))}$

Model

Answer Prior: CLEVR

- Apply fully connected layer of size $2d$ and ReLU activation to image representation
- Project to space of all possible answers and apply softmax

Model

Question Decoder: Overview

- Generate questions left to right using teacher forcing
- Embed words independently of context
- Use multi-layer RNN with attention to model interactions between question and context
- Compute likelihood of next word

Model

Question Decoder: Overview

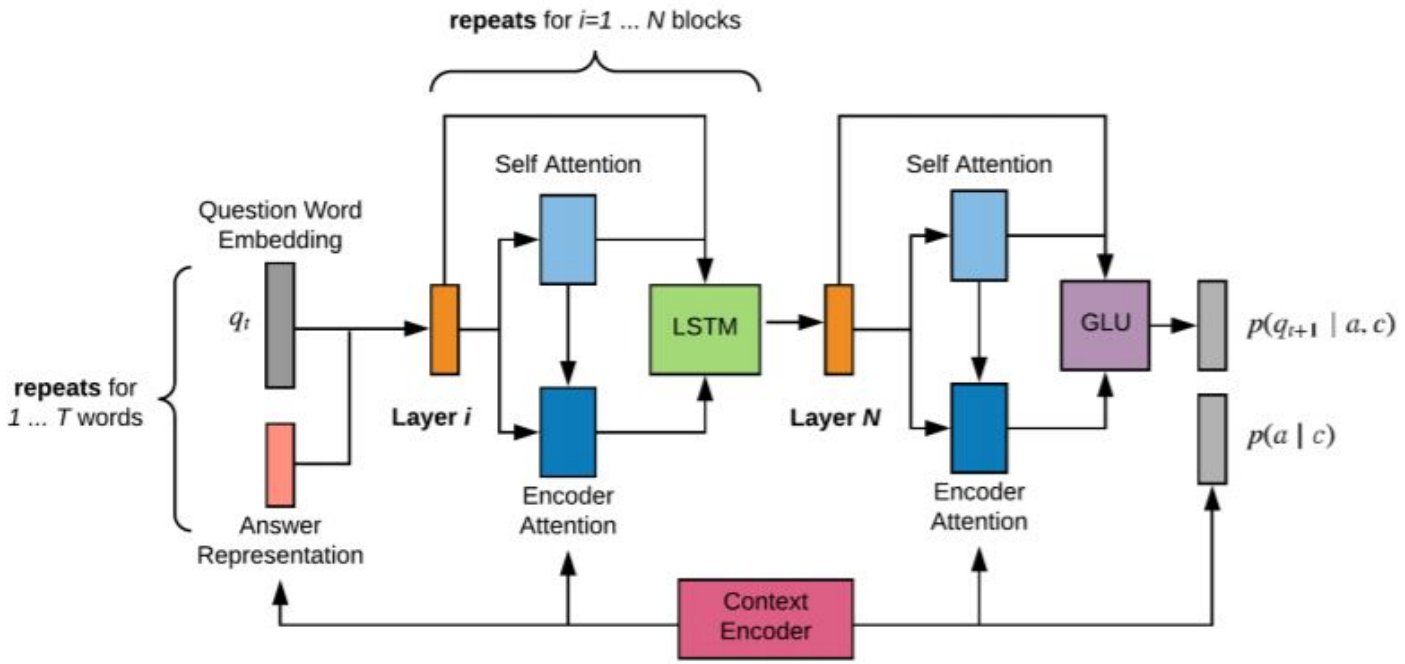


Figure 3: Architecture of GQA decoder. Multiple inputs to a layer indicates concatenation. Blocks after the first are connected with an additional residual connection, and LSTM cells also receive their state at time $t-1$ as an input.

Model

Question Decoder: Input Word Embeddings

- To represent SQuAD question words, use pretrained left to right language model (uni-directional version of ELMo)
- Then, use trainable LSTM layer of size d
- For CLEVR, just train embeddings of size d

Model

Question Decoder: Decoder Blocks

- Composed of a question self-attention layer and question-to-context attention mechanism; combined and fed into LSTM
- Context attention query is computed using both previous layer state and self-attention value
- Blocks connected with residual connections

Model

Question Decoder: Decoder Blocks

- Mechanisms similar to Vaswani et al. (2017), except use single-headed attention with bias term b_j for every context position j
- b_j is dot product of shared trainable vector and context encoding c_j; used to filter out irrelevant parts of context
- Final block combines self-attention and question-to-document attention using Gated Linear Unit layer to output vector of size d
- GLU uses gating to select relevant info for predicting next word

Model

Question Decoder: Output Word Probabilities

- Rare words more challenging for generative models; easier to guess from morphological clues than to generate them
- CLEVR has limited vocab, so just use unknown tokens; SQuAD has many rare words because of specialized vocab in wikipedia
- Use character-based softmax and copy mechanism

Model

Question Decoder: Output Word Probabilities, Character Softmax

- Infinite space of possible output words
- Approximate by normalizing over union of list of frequent words and any words which appear in any question or any document in batch
- Build character-based representation for each candidate word using pre-trained character CNN
- Add trainable linear projection to vector of size d
- Combine word and character based representations by summation

Model

Question Decoder:Output Word Probabilities, Pointer Mechanism

- Similar to Vinyals et al. (2015), use pointer mechanism to improve likelihood of specialized vocab in SQuAD by copying article words
- From final hidden state h_t, model chooses to copy based on classifier $p^{copy}(h) = \sigma(w^{copy} \cdot h)$, where $w^{copy} \in \mathbb{R}^d$ is trainable
- Model interpolates between generating with softmax $p^{gen}(h)$ and copying context word c_i
- Done using question-to-context attention probability from final layer α_t^i

$$p(q_t|q_{0:t-1}, c, a) = p^{copy}(h_t)\Sigma_i \alpha_t^i \mathbb{1}_{c_i=q_t} + (1 - p^{copy}(h_t))p^{gen}(q_t|h_t)$$

Model

Fine Tuning

- Using teacher forcing means model is not exposed to negative combinations of questions and answers during training
- Model can overfit as language model on questions, ignoring answer dependencies
- Thus, fine-tune model to make question more likely under the gold answer than other plausible answers
- Minimize $-\log \frac{p(q|a,c)p(a|c)}{\Sigma_{a' \in A} p(q|a',c)p(a'|c)}$, where A is 100 most likely answers from $p(a|c)$
- Only using this in loss produces poor results

Model

Inference

- Return answer $a*$ maximizing $a* = \text{argmax}_a p(q|a,c)p(a|c)$, which requires evaluating likelihood of question under each possible answer
- Use beam search to handle answer quantity
- Evaluate $p(a|c)$ for all possible answer spans up to length 30, taking top 250 candidates
- Only evaluate $p(q|a,c)$ for these top 250
- Correct answer is in beam for 98.5% of validation questions, suggesting it is not a major source of errors

Experiments

Large-scale Reading Comprehension

- Evaluate GQA on SQuAD dataset
- Competitive with state-of-the-art discriminative models
- Ensembles, data augmentation, and RL have better results, but these could be applied to GQA as well

Single Model	**Development**		**Test**	
	EM	F1	EM	F1
RaSOR (Lee et al., 2016)	66.4	74.9	67.4	75.5
BiDAF (Seo et al., 2016)	67.7	77.3	68.0	77.3
DrQA (Chen et al., 2017)	69.5	78.8	70.7	79.3
R-Net (Wang et al., 2017)	71.1	79.5	72.3	80.7
Weaver (Raison et al., 2018)	74.1	82.4	74.4	82.8
DCN+ (Xiong et al., 2017)	74.5	83.1	75.1	83.1
QANet + data augmentation x3 (Yu et al., 2018)	75.1	83.8	76.2	84.6
BiDAF + Self Attention + ELMo (Peters et al., 2018)	-	85.6	78.6	85.8
Reinforced Mnemonic Reader (Hu et al., 2018)	78.9	86.3	79.5	86.6
GQA	76.8	83.7	77.1	83.9

Table 1: Exact Match (EM) and F1 on SQuAD, comparing to the best published single models.

Experiments

Large-scale Reading Comprehension

- Ablation studies highlight importance of character-based softmax and pointer mechanism for rare words
- Fine tuning with discriminative objective improves results, but training with discriminative from scratch has weak performance

Single Model	Exact Match	F1
GQA	76.8	83.7
GQA (no fine-tuning)	72.3	80.1
GQA (no generative training)	64.5	72.2
GQA (no character-based softmax)	74.3	81.4
GQA (no pointer mechanism)	71.9	79.7
GQA (no answer-dependent context representation)	72.2	79.7
GQA (answer prior only)	13.4	16.1

Table 2: Development results on SQuAD for model ablations.

Experiments

Large-scale Reading Comprehension

- More answer candidates increases performance, but also computational cost

# Answer Candidates	Exact Match	F1
250	76.8	83.7
200	76.6	83.4
100	76.2	83.1
50	74.6	81.4
10	55.7	61.4

Table 3: Development results on SQuAD, varying the beam size during inference.

Experiments

Multihop Reasoning

- Evaluate CLEVR dataset, which tests visual reasoning
- GQA achieves 97.7% accuracy
- Could integrate MAC or FiLM to improve results

Single Model	Overall	Count	Exist	Compare Numbers	Query Attribute	Compare Attribute
Human	92.6	86.7	96.6	86.5	95.0	96.0
CNN+LSTM	52.3	43.7	65.2	67.1	49.3	53.0
CNN+LSTM+SA	76.6	64.4	82.7	77.4	82.6	75.4
CNN+LSTM+RN	95.5	90.1	97.8	93.6	97.9	97.1
CNN+GRU+FiLM	97.6	94.3	99.3	93.4	99.3	99.3
MAC	98.9	97.1	99.5	99.1	99.5	99.5
GQA	97.7	94.9	98.3	97.0	99.2	99.2

Table 4: Test results on CLEVR, demonstrating high accuracy at complex reasoning. GQA is the first approach to achieve high performance on both CLEVR and broad coverage QA tasks.

Experiments

Multihop Reasoning

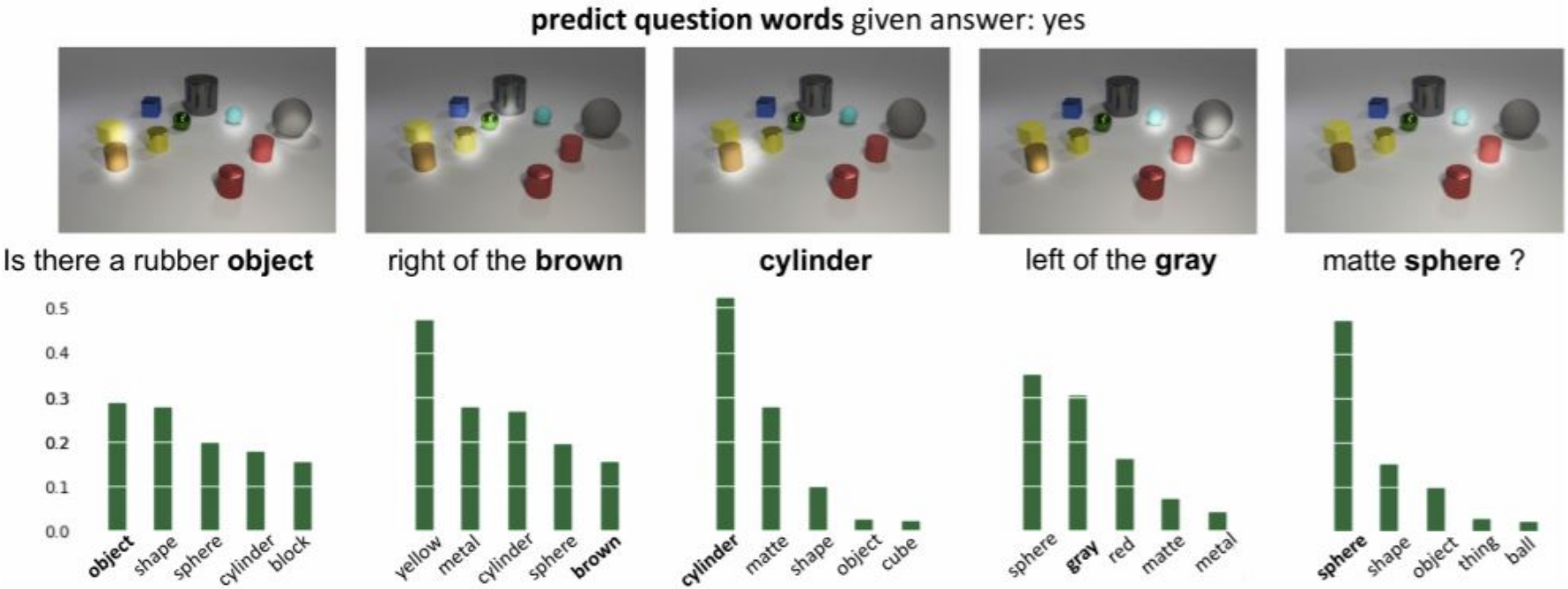


Figure 4: Final layer attention maps and word probabilities during question generation on a CLEVR validation question, when predicting the highlighted word. **(1)** The model considers all the rubber objects for predicting the next word. **(2)** Objects to the left of a rubber object are considered. **(3)** It describes the brown cylinder. **(4, 5)** Word distributions show the model understands the next words, but interestingly its attention focuses on the set of two objects meeting the constraints. In all cases, the word probability distributions are heavily skewed towards semantically valid choices.

Experiments

Learning from Biased Data

- Many popular QA datasets have biases which models can exploit
- For example, "when" questions with contexts that only have 1 date
- Create deliberately biased subsets of SQuAD based on named entity types (numbers, dates, people)
- Train on datasets where named entity type only appears once, validate on datasets with multiple appearances

Experiments

Learning from Biased Dat

Single Model	Numbers		Dates		People	
	EM	F1	EM	F1	EM	F1
Random Selection	19.37	26.18	19.37	26.18	19.37	26.18
First Occurrence	29.36	35.11	34.64	42.08	26.38	32.26
BiDAF	33.02	42.14	35.41	43.83	30.05	37.28
QANet	31.99	40.58	39.98	47.82	30.26	38.56
GQA (answer prior only)	37.15	45.54	35.55	43.85	32.56	38.79
GQA	58.49	67.56	64.71	72.51	53.09	61.93

Table 5: Exact Match (EM) and F1 on biased subsets of SQuAD. All answers in each subset have the indicated named-entity type; training documents have only one answer with this type, but for testing there are multiple plausible answers. Discriminative models perform comparably to question-agnostic baselines, whereas our generative model learns to generalise.

Experiments

Adversarial Evaluation

- Add distractor sentence to paragraph which almost answers question for each SQuAD example

Single Model	ADDSENT	ADDONESENT
BiDAF (Seo et al., 2016)	34.3	45.7
RaSOR (Lee et al., 2016)	39.5	49.5
MPCM (Wang et al., 2016)	40.3	50.0
ReasoNet (Shen et al., 2017)	39.4	50.3
Reinforced Mnemonic Reader (Hu et al., 2018)	46.6	56.0
QANet (Yu et al., 2018)	45.2	55.7
GQA	47.3	57.8

Table 6: F1 scores on ADVERSARIALSQUAD, which demonstrate that our generative QA model is substantially more robust to this adversary than previous work, likely because the additional adversarial context sentence cannot explain all the question words.

Experiments

Adversarial Evaluation

What we now call gravity was not identified as a universal force until the work of Isaac Newton. [...] Galileo was instrumental in describing the characteristics of falling objects [...] this acceleration due to gravity towards the surface of the Earth is usually designated as and has a magnitude of about 9.81 meters per second squared [...], and points toward the center of the Earth. [...] **Distractor:** Object falls about 5 times faster on Mars.

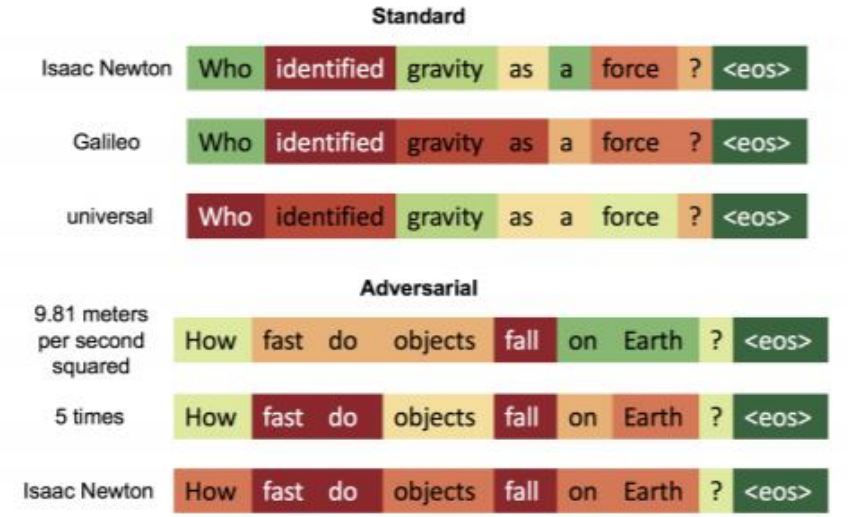


Figure 2: Probabilities of generating question words given different answers for a standard and an adversarial SQuAD question, allowing us to interpret which questions words are explained by the answer. In the standard setting, the model places greater probability on the question words that appear near *Isaac Newton*, such as *force* compared to *Galileo*. In the adversarial setting, the question word *Earth* distinguishes the true answer from the distractor.

Experiments

Long Context QA

- Extend to more challenging multi-paragraph setting
- Still train with single paragraph, but test on multi-paragraph
- Use multi-paragraph SQuAD: question matched with wikipedia article as context
- For each question, calculate $p(q|a)s(a, c)$ for proposed answer span a; $s(a, c)$ is logics from answer prior classifier
- Select answer which maximizes

Experiments

Long Context QA

Single Model	EM	F1
DrQA* trained on paragraph	59.1	67.0
Weaver trained on paragraph	60.6	69.7
DrQA* trained on documents	64.7	73.2
Weaver trained on documents	67.0	75.9
GQA trained on paragraph	71.4	78.4

Table 7: F1 scores on full document evaluation for SQuAD, which show our generative QA model is capable of selecting the correct paragraph for question answering even when presented with other similar paragraphs. Baselines are from (Raison et al., 2018).

Conclusion

- Introduced generative model for QA, leveraging the greater amount of information in questions compared to answers
- Better robustness to biased training data and adversarial testing data than state-of-the-art
- Future work could include combining information about an entity from multiple sources to generate questions