Scribe Note: Generative Question Answering: Learning to Answer the Whole Question

Presenter: Bill Zhang, Scribe: Ji Gao

June 1, 2019

1 Research Question

- Research problem: QA tasks on image or text based on context.
- QA models can learn on only partly of the question, which leads to bias and overfitting. The authors believe it's the discriminative style of learning caused this problem.
- **Method:** Define a generative objective function: Model $P(q|a,c)$ instead of $P(a|q,c)$. (q stands for question, a stands for the answer,and c stands for context).

2 Overview Figure

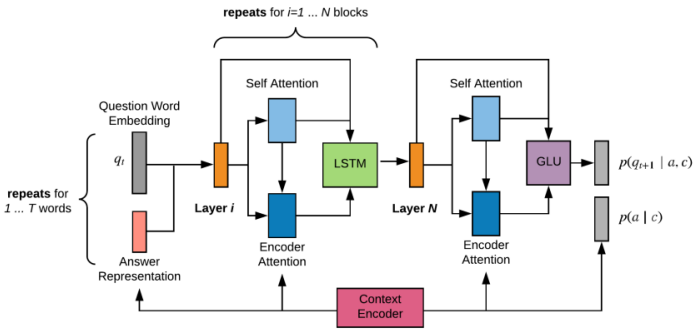

Figure 3: Architecture of GQA decoder. Multiple inputs to a layer indicates concatenation. Blocks after the first are connected with an additional residual connection, and LSTM cells also receive their state at time $t-1$ as an input.

3 Method

- **Overview:**
  - Use a context encoder to get an embedding of the context.
  - Directly infer a prior $P(a|c)$ on the answer.
  - Use a attention based recurrent model to create a decoder that models the probability of generating next word in the question.
  - Following chain rule, the loss function is defined as:

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

- **Encoder:** Encoder is treated differently for text and image data.
  - Text encoder
    - The text encoder of the context is similar to ELMo, using a character-based word embeddings and a 2-layer LSTM language model in forward and reverse directions.
    - The text encoder of the answer is a weighted sum of the word representation.
    - In SQuAD data, the authors calculate element-wise product of the word representation to the answer encoding.
  - Image encoder

    Image encoder is a pre-trained ResNet.

- **Decoder:** The output of the encoder is a context embedding vector. Decoder takes the context embedding as input and returns probability as output.
  - Answer decoder:

    Answer decoder returns $p(q, a|c)$, which is calculated from the
  - Question decoder:

    Question decoder is trained using a N block LSTM with additional attention.
    - Input word embedding is a pretrained uni-directional LSTM.
    - Decoder block is a self-attention LSTM (GLU, to be specifically) conmbined with an additional context attention.
      The attention is a single-headed attention. A bias term is added to the query-key score, which is calculated as a dot product of a shared trainable vector and context encoding.
    - Use different output layer. For the CLEVR, a word softmax layer is used for the small vocabulary. For SQuAD, the model first choose whether to copy using a pointer classifier, and then use a question-to-context attention probability.

– Fine tuning: Using another loss function for fine tuning, which only optimize the result on specific question answer pair, not on the language model.

$$-\log \frac{p(q|a,c)p(a|c)}{\sum_{a' \in A} p(q|a',c)p(a'|c)} \tag{2}$$

– Inference: Use beam search to infer the result.

$$a^* = \underset{a}{\operatorname{argmax}}\, p(q|a,c)p(a|c) \tag{3}$$

It only consider top 250 of the $p(a|c)$ to reduce computation cost.

4 Experiment

4.1 SQuAD result

SQuAD result is displayed in the tables. GQA get competitive result as a single model.

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.

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.

4.2 CLEVR result

CLEVR result is displayed. GQA get competitive result as a single model.

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.

4.3 Adversarial result

GQA appears to be robust to adversarial attacks, shown in the result of adversarial SQuAD.

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.

4.4 Long context

GQA shows good performance when context is longer.

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).