

EEE-QA: Exploring Effective and Efficient Question-Answer Representations

Zhanghao Hu*

Yijun Yang*

Junjie Xu*

Yifu Qiu

Pinzhen Chen

School of Informatics, University of Edinburgh

huzh666295@gmail.com, thomasyyj@outlook.com, smyjx1@163.com, yifu.qiu@ed.ac.uk, pchen3@ed.ac.uk

*Equal contribution

Overview

• Knowledge base question answering

- We challenged the existing question-answer encoding convention and explored finer representations
- **Semantic modelling:**
Using max pooling instead of CLS for QA scoring
- **Inference efficiency:**
Scoring a question and all answers in one go instead of scoring each QA pair

From One to Multiple Choices

• Previous 1AnP encoding

- Question in discrete tokens: $Q = [x_1, x_2, \dots, x_{|Q|}]$
- A set of n answer choices: $\mathcal{A} = \{A_1, A_2, \dots, A_n\}$
- Encode the question and each A_i as a pair $(Q, A_i)_{A_i \in \mathcal{A}}$:
 $\mathcal{H} = [H_{\langle s \rangle}, H_{x_1}, \dots, H_{x_{|Q|}}, H_{\langle |s \rangle_1}, H_{\langle |s \rangle_2}, H_{A_i}, H_{\langle |s \rangle_3}] =$
 $Encoder([\langle s \rangle, Q, \langle |s \rangle_1, \langle |s \rangle_2, A_i, \langle |s \rangle_3])$, where $\langle s \rangle$ and $\langle |s \rangle$
denote begin-of-sentence (or CLS) and end-of-sentence (or SEP)
tokens respectively

• Our nAnP encoding

- Append all candidates to the question and embed with Encoder () in one pass:
 $[\langle s \rangle, Q, \langle |s \rangle_1, \langle |s \rangle_2, A_1, \langle |s \rangle_3, A_2, \dots, A_n, \langle |s \rangle_{n+2}]$.
- Run pooling on each answer and compute multi-head attention between pooling outcomes for comparison
- Concatenate the pooling outcome of the question and each answer's final representation for scoring (Q, A) pairs.

Improved Question Representations

• Pooling

- Various pooling methods achieved higher performance than using CLS, but it is under-explored in the KGQA field.

• Commonsense QA results

System	Accuracy (std.)
RoBERTa-large	68.69 (± 0.56)
QA-GNN	73.41 (± 0.92)
JointLK	74.43 (± 0.83)
ACENet	74.72 (± 0.70)
GreaseLM	74.20 (± 0.40)
GreaseLM (our re-run)	73.57 (± 0.08)
+ mean pooling	73.73 (± 0.29)
+ max pooling	75.42 (± 0.52)
+ attentive pooling	73.97 (± 0.51)
+ layerwise CLS pooling	73.97 (± 0.16)

Table: Performance (accuracy, %) of our pooling investigation compared with previous works on CommonsenseQA.

Efficient Inference

• Performance of our systems on two dataset

- We test the performance on pure LM, KG+LM and interaction-enabled KG+LM task.
- When seeking improved memory efficiency, the baseline CLS pooling accuracies on both CommonsenseQA and OpenBookQA are sacrificed to a slight degree.
- However, this can be mitigated through our proposed techniques: max pooling as well as the gated answer representation mechanism. We observe on par if not higher performance when these are added, highlighting the effectiveness of our explicit inter-answer interaction.

System	Pooling	CommonsenseQA			OpenBookQA		
		PLM	+ KG	+ KG + Int	PLM	+ KG	+ KG + Int
1AnP	CLS pool	70.02	72.82	73.97	80.20	81.80	81.60
	Max pool	70.51	73.41	75.42	82.40	82.40	82.60
nAnP	CLS pool	67.12	69.62	68.82	79.40	78.40	81.80
	Max pool	67.12	68.90	69.14	79.40	82.60	82.20
nA1P	CLS pool	67.77	69.30	69.38	78.80	79.40	80.20
	Max pool	68.25	68.65	70.91	79.00	80.60	80.40
	Max pool + Gate	69.62	71.88	70.91	79.60	80.60	81.40

Table: Performance (accuracy, %) of our systems on CommonsenseQA and OpenBookQA.

Throughput

• Memory efficiency

- Our systems allow for larger batch sizes.
- The encoding strategy and comparison mechanism used in our approach lead to a slight compromise in processing time but better performance for a single pass through the model.
- The overall improvement in batch size significantly enhances throughput when QA needs to be performed at a large scale.

GPU Model	Mem. (GB)	Batch size (\uparrow)		Infer. time (\downarrow)	
		1AnP	nA1P ($\Delta\%$)	1AnP	nA1P ($\Delta\%$)
RTX A5000	24	100	160 (+60%)	4.61s	3.31s (-28%)
RTX 3090	24	100	160 (+60%)	2.45s	1.82s (-26%)
RTX 2080 Ti	11	30	45 (+50%)	1.13s	0.76s (-33%)
GTX 1080 Ti	11	30	45 (+50%)	2.64s	1.27s (-52%)
Titan X Pascal	12	40	55 (+38%)	3.56s	1.55s (-56%)
GTX 1080	8	10	20 (+100%)	2.21s	0.77s (-65%)

Table: An efficiency comparison between 1AnP and our nA1P: usable batch size and total inference time when solving 1000 QA on a single Nvidia GPU.

Conclusions

- Pooling produces better representations than [CLS].
- We propose a gated single-pass inference approach to encourage answer interactions and enhance efficiency.
- Experiments demonstrate (1) substantial gains with max pooling, surpassing state-of-the-art KBQA models. (2) We maintain a similar performance to the baseline while incurring less computation by 26-65% across various GPUs.

