Appendix for Paper "Multi-Task Learning for Conversational Question Answering over a Large-Scale Knowledge Base"

Tao Shen[1]*, **Xiubo Geng**[2], **Tao Qin**[2], **Daya Guo**[3], **Duyu Tang**[2], **Nan Duan**[2], **Guodong Long**[1] **and Daxin Jiang**[2]
[1]Centre for AI, School of Computer Science, FEIT, University of Technology Sydney
[2]Microsoft, Beijing, China
[3]The School of Data and Computer Science, Sun Yat-sen University
tao.shen@student.uts.edu.au, guodong.long@uts.edu.au
{xiubo.geng,taoqin,dutang,nanduan,djiang}@microsoft.com
guody5@mail2.sysu.edu.cn

A Model Details

A.1 Word Embedding

Given an user question sentence U, a tokenizing method (e.g., punctuation or wordpiece tokenizer (Wu et al., 2016)) is applied to the sentence for a list of tokens, i.e., $\boldsymbol{U} = [\boldsymbol{u_1}, \cdots, \boldsymbol{u_{n-1}}, \boldsymbol{u'}]$, where $\boldsymbol{u_i}$ or $\boldsymbol{u'}$ is an one-hot vector whose dimension equals to distinct tokens N in vocabulary, and n is the length of $\boldsymbol{U}$. Note that a special token $\boldsymbol{u'}$ is appended to the tokenized sentence, corresponding to the token *[CTX]*. Then, randomly initialized or pre-trained (Mikolov et al., 2013; Pennington et al., 2014) embeddings are applied to $\boldsymbol{U}$ and thus transform discrete tokens to a sequence of low-dimension distributed embeddings, i.e., $\boldsymbol{X} = [\boldsymbol{x_1}, \boldsymbol{x_2}, ..., \boldsymbol{x_n}] \in \mathbb{R}^{d_e \times n}$ where d_e is embedding size. This process is formulated as $\boldsymbol{X} = \boldsymbol{W^{(enc)}U}$ where $\boldsymbol{W^{(enc)}} \in \mathbb{R}^{d_e \times N}$ is the trainable word embedding weight matrix.

A.2 Pointer-equipped Semantic Parsing

A.2.1 Encoder of Seq2seq Model

To model contextual dependencies between tokens and generate context-aware representations, we leverage stacked two-layer multi-head attention mechanism with additive positional encoding (Vaswani et al., 2017). The stacking scheme is identical to that in (Vaswani et al., 2017): two-layer feed forward network with activation function (FFN) follows each multi-head attention, and residual connection (He et al., 2016) with layer normalization (Lei Ba et al., 2016) is applied. This process is briefly denoted as

$$\boldsymbol{H} = [\boldsymbol{h_1}, \cdots, \boldsymbol{h_n}] \triangleq \boldsymbol{X'} \in \mathbb{R}^{d_e \times n}, \text{where,} \quad (1)$$

$$^{2\times}[\boldsymbol{X'} = \text{FFN}(\text{MultiHead}(\boldsymbol{X'}, \boldsymbol{X'}, \boldsymbol{X'}))], \quad (2)$$

$$\boldsymbol{X'} = \boldsymbol{X} + \boldsymbol{W^{(pe)}}, \quad (3)$$

where $\boldsymbol{H}$ is a sequence of contextual embeddings, $\boldsymbol{W^{(pe)}} \in \mathbb{R}^{d_e \times n}$ is learnable weights of PE and the three arguments for MultiHead are *value*, *key*, *query* for an attention mechanism.

A.2.2 Decoder of Seq2seq Model

Similar to token embedding in encoder (§A.1), we embed the j-th decoder input token as $\boldsymbol{z_j}$ via a randomly initialized embedding weight matrix $\boldsymbol{W^{(dec)}} \in \mathbb{R}^{d_e \times |\mathbb{V}^{(dec)}|}$. We use $\boldsymbol{Z} = [\boldsymbol{z_1}, \cdots, \boldsymbol{z_m}] \in \mathbb{R}^{d_e \times m}$ to represent all tokens in a gold logical form sketch, where m denotes the length of gold sketch.

The basic structure of proposed logical form decoder is same as that in the original Transformer (Vaswani et al., 2017) except only two stacked layers are used here. Each layer of the decoder is bottom-up comprised of self-attention with forward mask, cross attention between decoder and encoder and FFN, which we briefly formulate as

$$\boldsymbol{S} = [\boldsymbol{s_1}, \cdots, \boldsymbol{s_m}] \triangleq \boldsymbol{Z} \in \mathbb{R}^{d_e \times m}, \text{where,} \quad (4)$$

$$^{2\times}[\boldsymbol{Z} = \text{FFN}(\text{MultiHead}(\quad (5)$$
$$\boldsymbol{H}, \boldsymbol{H}, \text{MultiHead}^{mask}(\boldsymbol{Z}, \boldsymbol{Z}, \boldsymbol{Z})))].$$

where $\boldsymbol{S}$ is a sequence of decoding hidden states.

A.3 Multi-task Learning

We propose to employ a multi-task learning strategy to learn a entity detection (ED) model jointly with the pointer-equipped semantic parsing model because the supervision information from ED, i.e, IOB tagging, can provide all entities spans in the input question, which thus results in better performance than separate learning.

The reasons why we use a multi-task learning to jointly learn the semantic parsing model and ED rather than directly equip the semantic parsing model with span prediction (Seo et al., 2017)

* Work done while the author was an intern at Microsoft.

are that 1) the supervision information of the entities not existing in the gold logical form but appearing in the question is lost; 2) deeper network is required when predicting the end index of the target as shown in (Seo et al., 2017) and 3) the well-solved entity detection method can provide correction for the pointer even with slight deviation during inference phrase, in contrast, span-based model usually leads to error aggregation.

A.4 Inverted Index

Based on each entity text in Wikidata, we traversed its substring whose length is not less than that of its full text minus a threshold, and then, we separately calculated Levenshtein Distance between the full text and each substring as a score for the map from the substring to corresponding full text. Since multiple entities could generate identical substring, we kept maps with largest scores and used the maps to build a dictionary for future queries.

B Supplemental Experiment Results

B.1 Precision and Recall for Main Paper

Since we report the F1 score for brief demonstration in the main paper, in this section, we report the corresponding recall and precision detailedly: 1) as shown in Table 7, the results of the proposed model compared with baselines are presented; 2) as shown in Table 8, the ablation study is presented; and 3) as shown in Table 9, the performance improvement comparison after sophisticated strategies applied is provided.

B.2 Comparison to D2A

Question Type	D2A	Ours
Simple Question (Direct)	2.6	1.5
Clarification	2.7	1.4
Simple Question (Coreferenced)	2.7	1.4
Quantitative Reasoning (Count) (All)	2.9	1.5
Logical Reasoning (All)	2.7	1.6
Simple Question (Ellipsis)	2.6	1.6
Verification (Boolean) (All)	2.8	1.4
Quantitative Reasoning (All)	2.7	1.4
Comparative Reasoning (Count) (All)	2.8	1.4
Comparative Reasoning (All)	3.0	1.4
Overall	2.9	1.5

Table 1: The averaged number of entity candidates from entity linking.

To further demonstrate that the proposed model is superior to the previous D2A model in term of entity linking and logical form generation, we conduct the following comparisons.

First, as shown in Table 1, the average number of entity candidates in test set from entity linking of the proposed model is $2\times$ less than that of D2A, which means the proposed approach provides the downstream subtask with more accurate entity linking results.

Question Type	D2A	Ours
Simple Question (Direct)	0.8960	0.9520
Clarification	0.8281	0.9323
Simple Question (Coreferenced)	0.8177	0.8952
Quantitative Reasoning (Count) (All)	0.8385	0.9581
Logical Reasoning (All)	0.8726	0.9791
Simple Question (Ellipsis)	0.9364	0.9474
Verification (Boolean) (All)	0.7448	0.9637
Quantitative Reasoning (All)	0.9304	0.9832
Comparative Reasoning (Count) (All)	0.8165	0.9863
Comparative Reasoning (All)	0.8312	0.9727
Overall	0.8499	0.9475

Table 2: Ratio of non-empty logical form.

Second, we compare the proposed model with D2A in term of logical form generation where the logical form would be empty due to timeout or illegal logical forms during beam search. As demonstrated in Table 2, the proposed model obtains less ratio of empty logical form than D2A.

Question Type	D2A	Ours	+BERT
Simple Question (Direct)	0.7967	0.8519	0.8664
Clarification	0.2385	0.6408	0.6414
Simple Question (Coreferenced)	0.5341	0.7234	0.7469
Quantitative Reasoning (Count) (All)	0.5000	0.6947	0.7004
Logical Reasoning (All)	0.3692	0.0791	0.3196
Simple Question (Ellipsis)	0.7533	0.8843	0.8878
Verification (Boolean) (All)	0.1757	0.5278	0.5854
Quantitative Reasoning (All)	0.8913	0.9792	0.9911
Comparative Reasoning (Count) (All)	0.3235	0.8924	0.9121
Comparative Reasoning (All)	0.2483	0.9053	0.9242
Overall	0.5522	0.7167	0.7546

Table 3: accuracy of entities in predicted logical form.

Third, we list the accuracies of the entities appearing in the predicted logical form for D2A, our standard approach and BERT-based model, which verifies that the proposed approach can significantly improve the performance of entity linking during entity detection and entity prediction during logical form generation. Note that the analysis for performance reduction of *Logical Reasoning (All)* is elaborated in the main paper.

B.3 Multi-task Learning

The multi-task learning framework increases the accuracy of logical form generation while keeping

a satisfactory performance of entity detection, and consequently improves the final question answering task via logical form execution. In this section, we detailedly list all metrics to measure the performance for both two subtasks in the case of our approach with or without multi-task learning. To evaluate the logical form generation, we also apply BFS method to test set for gold logical form (inevitably existing spurious ones).

Question Type	**Ours**	**w/o Multi**
Comparative Reasoning (All)	0.1885	0.1885
Logical Reasoning (All)	0.6256	0.6188
Quantitative Reasoning (All)	0.6403	0.6188
Simple Question (Coreferenced)	0.8721	0.8663
Simple Question (Direct)	0.8772	0.8715
Simple Question (Ellipsis)	0.9073	0.9034
Comparative Reasoning (Count) (All)	0.1601	0.1495
Quantitative Reasoning (Count) (All)	0.5711	0.5564
Verification (Boolean) (All)	0.7638	0.7565
Overall	0.7940	0.7872

Table 4: Sketch accuracy for logical form generation.

		Ours	w/o Multi
IOB Tagging	Accuracy	0.9967	0.9975
	F1 Score	0.9941	0.9955
	Precision	0.9960	0.9972
	Recall	0.9923	0.9938
Entity Type	Accuracy	0.9822	0.9844
	F1 Score	0.9674	0.9717
	Precision	0.9958	0.9971
	Recall	0.9407	0.9475

Table 5: Performance of IOB tagging and entity type prediction.

As shown in Table 4 and 5, the model with multi-task learning can outperform that without multi-task learning in term of logical form generation from semantic parsing model. And, although ~ 0.002 performance reduction is observed for entity detection subtask, the performance of entity detection and linking is good enough for the downstream task, which thus poses a very minor effect on the performance of KB-QA.

B.4 BFS Success Ratio

Given the final answer to a question as well as gold entities, predicates and types, we conduct a BFS method to search the gold logical form, which may result in search failure due to limited time and buffer. We list the success ratio of BFS for training data of CSQA in Table 6.

Question Type	**#Example**	Ratio
Simple Question (Direct)	274527	0.96
Simple Question (Ellipsis)	34549	0.97
Quantitative Reasoning (All)	58976	0.46
Quantitative Reasoning (Count) (All)	114074	0.67
Logical Reasoning (All)	66161	0.61
Simple Question (Coreferenced)	173765	0.86
Verification (Boolean) (All)	77167	0.75
Comparative Reasoning (Count) (All)	59557	0.37
Comparative Reasoning (All)	57343	0.32

Table 6: The BFS search success ratio w.r.t. difference question type.

C Supplemental Analysis

We also observe that the improvement of MaSP over D2A for some question types is relatively small especially for logical reasoning questions. Furthermore, for logical reasoning, we find that the accuracy of entities in final logical forms is only 8%, and there are usually two distinct entities needed to produce a correct logical form. This means the presented shallow network, i.e., two-layer multi-head attention, cannot handle such complex cases. We study a case here for better understanding. Given, "*Which diseases are a sign of lead poisoning or pentachlorophenol exposure?*", D2A produces "(*union* (*find* {*lead poisoning*}, *symptoms*), (*pe...ol exposure*))" where entities are correct but operator is wrong, our approach produces "(*union* (*find* {*pe...ol exposure*}, *symptoms*), (*union* (*find* {*pe...ol exposure*}, *symptoms*))" where the entities are wrong, while our approach plus BERT (Devlin et al., 2018) as encoder can produce correct logical form that is "(*union* (*find* {*pe...ol exposure*}, *symptoms*), (*union* (*find* {*lead poisoning*}, *symptoms*))".

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*.

Methods		HRED+KVmem		D2A (Baseline)		Our Approach	
Question Type	**#Example**	Recall	Precision	Recall	Precision	Recall	Precision
Overall	-	18.40%	6.30%	66.83%	66.57%	78.07%	80.48%
Clarification	12k	25.09%	12.13%	37.24%	33.97%	84.18%	77.66%
Comparative Reasoning (All)	15k	2.11%	4.97%	44.14%	54.68%	59.83%	81.20%
Logical Reasoning (All)	22k	15.11%	5.75%	65.82%	68.86%	61.92%	78.00%
Quantitative Reasoning (All)	9k	0.91%	1.01%	52.74%	60.63%	69.14%	79.02%
Simple Question (Coreferenced)	55k	12.67%	5.09%	58.47%	56.94%	76.94%	76.01%
Simple Question (Direct)	82k	33.30%	8.58%	79.50%	77.37%	86.09%	84.29%
Simple Question (Ellipsis)	10k	17.30%	6.98%	84.67%	77.90%	85.50%	82.03%
Question Type	**#Example**	Accuracy		Accuracy		Accuracy	
Verification (Boolean)	27k	21.04%		45.05%		60.63%	
Quantitative Reasoning (Count)	24k	12.13%		40.94%		43.39%	
Comparative Reasoning (Count)	15k	8.67%		17.78%		22.26%	

Table 7: Results of comparisons for KB-QA with baselines.

Methods	Our Approach		w/o ET		w/o Multi		w/o Both	
Question Type	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision
Overall	78.07%	80.48%	68.78%	72.15%	75.75%	77.73%	66.75%	69.75%
Clarification	84.18%	77.66%	69.79%	66.32%	70.12%	62.88%	56.96%	52.51%
Comparative Reasoning (All)	59.83%	81.20%	57.48%	78.45%	53.62%	71.06%	50.86%	67.59%
Logical Reasoning (All)	61.92%	78.00%	54.43%	73.73%	61.04%	76.27%	54.16%	73.91%
Quantitative Reasoning (All)	69.14%	79.02%	69.14%	79.02%	60.86%	68.73%	60.86%	68.72%
Simple Question (Coreferenced)	76.94%	76.01	64.92%	64.96%	74.65%	74.06%	63.06%	63.24%
Simple Question (Direct)	86.09%	84.29%	75.87%	74.62%	85.88%	84.01%	75.84%	74.56%
Simple Question (Ellipsis)	85.50%	82.03%	80.12%	76.85%	84.28%	81.11%	78.96%	75.97%
Question Type	Accuracy		Accuracy		Accuracy		Accuracy	
Verification (Boolean)	60.63%		45.40%		60.43%		45.02%	
Quantitative Reasoning (Count)	43.39%		39.70%		37.84%		43.39%	
Comparative Reasoning (Count)	22.26%		19.08%		18.24%		22.26%	

Table 8: Ablation study. "w/o ET" stands for removing entity type prediction in Entity Detection; "w/o Multi" stands for learning two subtasks separately in our framework; and "w/o Both" stands for a combination of "w/o ET" and "w/o Multi".

Methods	Vanilla		w/ BERT		Larger Beam Size	
Question Type	Recall	Precision	Recall	Precision	Recall	Precision
Overall	78.07%	80.48%	79.67%	81.56%	80.39%	82.75%
Clarification	84.18%	77.66%	83.24%	76.01%	86.90%	80.11%
Comparative Reasoning (All)	59.83%	81.20%	58.79%	75.21%	60.25%	81.67%
Logical Reasoning (All)	61.92%	78.00%	72.56%	83.24%	62.16%	78.58%
Quantitative Reasoning (All)	69.14%	79.02%	66.91%	74.35%	69.14%	79.02%
Simple Question (Coreferenced)	76.94%	76.01%	78.05%	77.85%	79.54%	78.52%
Simple Question (Direct)	86.09%	84.29%	86.84%	85.96%	89.26%	87.33%
Simple Question (Ellipsis)	85.50%	82.03%	86.38%	83.32%	88.78%	85.22%
Question Type	Accuracy		Accuracy		Accuracy	
Verification (Boolean)	60.63%		63.85%		61.96%	
Quantitative Reasoning (Count)	43.39%		47.14%		44.22%	
Comparative Reasoning (Count)	22.26%		25.28%		22.70%	

Table 9: Comparisons with different experimental settings. "Vanilla" stands for standard settings of our framework. "w/ BERT" stands for incorporating BERT. "w/ Large Beam" stands for increasing beam search size from 4 to 8.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *ICLR*.

Ashish Vaswani, Shazeer, Noam, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.