Appendices

A Details of Experimental Settings

We tune hyper-parameters on the development set. For each base model, we first tune its hyper-parameters to achieve the best performance, and then we fix its best hyper-parameters before plugging PIECER. The criterion for selecting the best hyper-parameters is the development F1. Details of the general hyper-parameters and hyper-parameters for each base model are described as follows.

General hyper-parameters: (1) Empirically, we use AdamW with $\beta 1 = 0.9$, $\beta 2 = 0.98$, eps = 1e-6, weight decay = 1e-2 as the optimizer. (2) Empirically, we adopt a slanted triangular learning rate scheduler, which first linearly increases the learning rate from 0 to the peak value during the first 6% steps, and then linearly decreases it to 0 during the remaining steps. (3) Empirically, we apply exponential moving average with a decay rate of 0.9999 on trainable parameters. (4) Empirically, we set all dropout rates to 0.1. (5) We try the number of Highway GAT layers to in {1, 2, 3, 4, 5}, and finally select 3. (6) We try the number of attention heads in {1, 2, 4, 8}, and finally select 4.

Hyper-parameters for QANet: (1) We try the peak learning rate in {5e-4, 1e-3, 5e-3}, and finally select 1e-3. (2) We try the hidden dimension in {64, 128, 256}, and finally select 128. (3) We try the batch size in {8, 16, 32, 64}, and finally select 32. (4) We try to plug PIECER after the embedding layer, after the encoding layer, and at both these two positions, and finally select plugging at both two positions. (5) Since QANet has its own self-matching layer, we remove the optional self-matching submodule in PIECER. (6) We train for 30 epochs and evaluate the model on the development set after each epoch. Finally, we report the best F1 achieved during 30 epochs and use the corresponding model to predict answers on the test set.

Hyper-parameters for $\mathbf{BERT_{base}}$: (1) We try the peak learning rate for BERT module in {1e-5, 3e-5, 5e-5}, and finally select 1e-5. (2) We try the peak learning rate for other modules in {5e-4, 1e-3, 5e-3}, and finally select 5e-4. (3) We set the hidden dimension to 768, the same as $\text{BERT}_{\text{base}}$. (4) We try the batch size in {4, 8, 16, 32}, and finally select 4. (5) We plug PIECER between $\text{BERT}_{\text{base}}$ and the predicting layer since $\text{BERT}_{\text{base}}$ is impartible. (6) We keep the optional self-matching submodule in PIECER. (7) We train for 4 epochs and evaluate the model on the development set after each epoch. Finally, we report the best F1 achieved during 4 epochs and use the corresponding model to predict answers on the test set.

Hyper-parameters for $\mathbf{BERT_{large}}$: (1) We try the peak learning rate for BERT module in {1e-5, 3e-5, 5e-5}, and finally select 3e-5. (2) We try the peak learning rate for other modules in {5e-4, 1e-3, 5e-3}, and finally select 1e-3. (3)

We set the hidden dimension to 1024, the same as $\text{BERT}_{\text{large}}$. (4) We try the batch size in {4, 8, 16, 32}, and finally select 32. (5) We plug PIECER between $\text{BERT}_{\text{large}}$ and the predicting layer since $\text{BERT}_{\text{large}}$ is impartible. (6) We keep the optional self-matching submodule in PIECER. (7) We train for 4 epochs and evaluate the model on the development set after each epoch. Finally, we report the best F1 achieved during 4 epochs and use the corresponding model to predict answers on the test set.

Hyper-parameters for $\text{RoBERTa}_{\text{base}}$: (1) We try the peak learning rate for RoBERTa module in {1e-5, 3e-5, 5e-5}, and finally select 1e-5. (2) We try the peak learning rate for other modules in {5e-4, 1e-3, 5e-3}, and finally select 5e-4. (3) We set the hidden dimension to 768, the same as $\text{RoBERTa}_{\text{base}}$. (4) We try the batch size in {4, 8, 16, 32}, and finally select 4. (5) We plug PIECER between $\text{RoBERTa}_{\text{base}}$ and the predicting layer since $\text{RoBERTa}_{\text{base}}$ is impartible. (6) We keep the optional self-matching submodule in PIECER. (7) We train for 4 epochs and evaluate the model on the development set after each epoch. Finally, we report the best F1 achieved during 4 epochs and use the corresponding model to predict answers on the test set.