D Technical Note on Modified System

This technical note describes the modifications made to the originally published system to make a faster and more accurate system. We have incorporated language modelling pre-training in our modules using GPT-2 small (Radford et al., 2019) for question generation and BERT (Devlin et al., 2019) for question answering. The official code and demo uses the modified version of the system.

D.1 Dataset

The primary modification in the dataset tackles the problem of coreferences in GENERAL questions, as described in Section 5.2. This is a common problem in QuAC and CoQA due to their contextual setup. We pass every question through the spaCy pipeline extension `neuralcoref`[20] using the paragraph context to resolve co-references. We have also black-listed a few more question templates (such as "*What happened in <year>?*") due to their unusually high prevalence in the dataset.

D.2 Question Generation

Our question generation system is now fine-tuned from a pretrained GPT-2 small model (Radford et al., 2019). Our modified system is based on Wolf et al. (2019) and uses their codebase[21] as a starting point.

We train our question generation model using the paragraph and answer as language modelling context. For GENERAL questions, our input sequence looks like "`<bos> ..paragraph text.. <answer-general> ..answer text.. <question-general> ..question text.. <eos>`" and equivalently for SPECIFIC questions. In addition, we leverage GPT-2's segment embeddings to denote the specificity of the answer and question. Each token in the input is assigned one out of five segment embeddings (paragraph, GENERAL answer, SPECIFIC answer, GENERAL question and SPECIFIC question). Finally, answer segment embeddings were used in place of paragraph segment embeddings at the location of the answer in the paragraph to denote the position of the answer in the paragraph. For an illustration, refer to Figure A2.

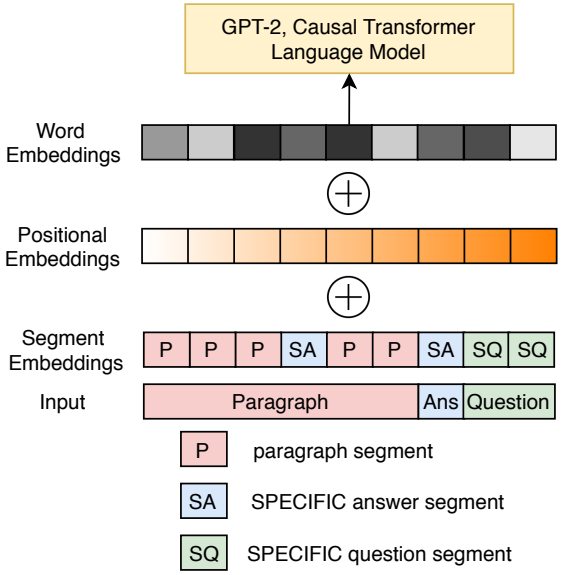

Figure A2: An illustration of the model used for generating a SPECIFIC question. A paragraph (context), answer and question and concatenated and the model is optimized to generate the question. Separate segment embeddings are used for paragraphs, GENERAL answers, GENERAL questions, SPECIFIC answers and SPECIFIC questions. Note that the answer segment embedding is also used within the paragraph segment to denote the location of the answer.

The question generation model now uses top-p nucleus sampling with $p = 0.9$ (Holtzman et al., 2019) instead of beam search and top-k sampling. Due to improved question generation quality, we no longer need to over-generate questions.

D.3 Question Answering

We have switched to a BERT-based question answering module (Devlin et al., 2019) which is trained on SQuAD 2.0 (Rajpurkar et al., 2018). We have used an open source PyTorch implementation to train this model[22].

D.4 Question Filtering

We have simplified the question filtering process to incorporate a simple QA budget (described in Section 7). Users are allowed to specify a custom "GENERAL fraction" and "SPECIFIC fraction" which denotes the fraction of GENERAL and SPECIFIC questions retained in the final output.

[20] https://github.com/huggingface/neuralcoref/

[21] https://github.com/huggingface/transfer-learning-conv-ai

[22] https://github.com/huggingface/pytorch-pretrained-BERT