Question Generation for Standardized English Exams

Group 10
Language Technology Institute
Carnegie Mellon University
Pittsburgh, PA 15213
{kchian, jiayueg, tongh, hcai2}@cs.cmu.edu

Abstract

Question generation is growing in popularity for its prospects in improving aspects of the education industry, question answering systems, and benchmarks in natural language understanding. Recently, the field has advanced from statistical and templating methods to attempting to generate sophisticated questions through technology such as seq-2-seq deep recurrent neural networks and enormous pre-trained transformer models. However, this is often limited to generating specific, simple questions from short paragraph snippets common in question answering datasets. Since real-world text data is less structured, our contribution is a system that can take arbitrarily long context and generate various questions from within. To generate complex, educational questions, we expand on the idea of using cause-and-effect labels as a starting point for question generation with a novel context selection portion of our pipeline. We evaluate both automatically and manually on how well we are able to craft questions similar to the Test of English as a Foreign Language (TOEFL) listening exam. Furthermore, our approach to preprocess arbitrarily long text for inputs to our generation model can be expanded to other branches of text generation.

1 Introduction

Neural Question Generation (QG) is a branch of natural language processing (NLP) that utilizes deep neural networks to generate questions. This extends work in automatic Question Generation (AQG) which has uses in augmenting Question Answering (QA) systems and creating educational material[25, 26]. Question Generation is challenging primarily due to the unstructured nature of text data, especially in the context of certain educational contexts, and a limited, subjective ground truth reference for evaluation.

To overcome these challenges, we perform a set of studies with two datasets with differing properties. The first dataset contains a relatively small amount of data from the TOEFL Listening section and contains full-length passages, multiple choice answers, and questions.[19, 20] The second dataset is called ReAding Comprehension Dataset From Examinations (RACE) [18]. RACE contains short passages and multiple choice questions geared towards middle school and high school English exams.

The goal of our project is to use QG/QA for standardized tests and foreign language learning material such that arbitrary passages can be used in an educational setting by teachers and students.The perhaps obvious motivation is to use our methods in an educational setting, allowing students of all backgrounds to effectively practice their English comprehension on questions similar to those used in standardized exams. It would enable schools and exam bootcamps to augment their ability to prepare students for an increasingly globalized world where English is prominent, without having to rely solely on expensive tutoring or meager resources on previous exams.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

We also publicize our code to promote further research in this area: `https://github.com/jiayueg/11785_project`

2 Related Work

2.1 Models for multiple choice QA/QG

With the proposal of attention mechanism and transformer architecture [1], many state-of-the-art architectures have been proposed using transformer model as building block. One of such model is BERT [2], which is built by stacking layers of transformer encoder. During training, BERT uses masked word prediction and next sentence prediction. Subsequently, Text-To-Text Transfer Transformer (T5) [3] was proposed, which is pre-trained on Colossal Clean Crawled Corpus (C4) dataset using fill-in-the-blank-style denoising objectives.[1] Such huge-scale pre-trained models can be very useful for multi-task learning in NLP, where people can just fine-tune the model based on specific task, which alleviates the problems like the scarcity of training data, and can usually achieve good result. To alleviate the problem of parameters explosion in such large scale transformer model, ALBERT model has been proposed[28], which allows parameter sharing across all BERT encoder layers and replaces the next sentence prediction objective in BERT with sentence order prediction objective.

For our specific TOEFL question QA-QG task, we separate the project into two different components: a question generation network based on T5 whose inputs are answer and context(passage), an answer generation network based on T5 whose input is question and context(passage). Additionally, a passage selection mechanism based on end-to-end memory network [30] and recurrent attention is introduced as an extra component to select the most relevant part of the passage.

There is other literature on attempts to generate questions for an educational context, and the idea is gaining traction [5, 6]. These groups generally chose to focus on using QG techniques on generic passages such as Wikipedia articles or textbook excerpts using the famous SQuAD dataset [8], or generating questions for the purpose of assessing students. In the past, people focused mostly on templating, parsing, and statistical approaches using computational linguistics for question generation [7, 22, 23]. These methods have limitations in generating complex, open ended questions across full passages. One particular method, info-HCVAE [31] approximates the conditional distribution of question given passage and answer and the conditional distribution of answer given passage using empirical risk lower bound. It then generates QA paired given the learned distribution. Our approach differs by honing in on one specific application of question generation for English learners, specifically using advancements in deep learning and transformers for targeting reading comprehension. Our task is also different than QA/QG task on SQuAD, since in SQuAD, the answer is contained directly in some marked span in the original passage; however, our model focuses on multiple choice QA/QG, where the span of answer is not masked in the passage. So the model needs to first understand the passage and to perform multi-hop reasoning on passage in order to generate legitimate answer and question.

3 Methodology

3.1 Model Description

3.1.1 Question Generation Model

The question generation network is also a fine-tuned T5 Conditional Generator model. We explore several inputs formats to our model. One approach selects blocks of sentences based on the BERTScore between sentences and the reference question; one approaches select sentences based on learnable End-To-End Memory Network [30]; we also decides whether or not to incorporate a given answer as an input. During inference, the aim is to have a model that will generate TOEFL-like questions from arbitrary passages, allowing educators and students to scale their learning material based on recent news or interests. Automatic evaluation will be centered around to what extent a model is able to recreate questions from a test set of TOEFL Questions.

[1] https://www.tensorflow.org/datasets/catalog/c4

3.1.2 Cause-and-effect Model

To enhance the performance of the question generation model, we used a pretrained cause-and-effect model from [24], which is generalized across different domains in natural language processing, so there is no need to fine-tune based on the TOEFL-QA and RACE dataset. The cause-and-effect model is pre-trained to extract causal relationships based on an input string, which represents the context (usually within a range of two sentences). The generated cause-effect relationship is then concatenated to the context that our QG model works on. The QG model is supposed to take advantage of this extra prior knowledge and improve the quality of the generated questions.

3.1.3 Question Answering Model

In addition to generating questions based on arbitrarily long contexts, we also try to answer them by a machine learning model. Our model for question answering is t5-small that is fine-tuned on the TOEFL-QA and RACE dataset. We feed the T5 model with tokenized questions and contexts, whose lengths are truncated to 396 and 32, respectively, and let it learn from the ground-truth answers of the datasets. The goal of building a QA model is to generate reasonably accurate answers for our QG model, allowing students to get feedback for their reading assignments, which helps them practice their language skills.

3.1.4 End-To-End Memory Network for Sentence Selection

One significant drawback of transformer model is that there is a fixed length limit on the input sequence during the tokenization process. To deal with this problem, we need to implement some selection mechanism that allows our model to focus on the most relevant part of the passage and ignore other.

In End-To-End Memory Network paper [30], the author proposes a recurrent attention network and several variations that will be able to perform soft sentence selection. The model performs attention in a recurrent manner and add residual connections among hops. The exact procedures is listed below:

1. Given a set of sentences $\{x_1, x_2, ..., x_i\}$, the model first converts the tokenized sentences to key vectors $\{k_i\}$ using a learnable embedding matrix A. It also uses another embedding matrix C to produce a set of value vectors $\{v_i\}$ for the sentences.
2. For each sentence value embedding v_i and sentence key embedding k_i, a positional embedding based on the length of the sentence is added. The positional embedding PE_j is a column vector calculated as below:

   $$PE_{kj} = (1 - j/J) - (k/d)(1 - 2 * j/J)$$

   where J represents the number of word in the embedding, and d represents the dimension of the embedding. The value and key embeddings are calculated as following:

   $$k_i = \sum_j PE_j * A(x_{ij})$$

   $$v_i = \sum_j PE_j * C(x_{ij})$$

3. For each question, an embedding matrix B is used to convert the tokenized question into question embedding q, which serves as query.
4. The attention score p_i is calculated as following:

   $$p_i = Softmax(q^T k_i)$$

   We then take a weighted sum over value vectors $\{v_i\}$ to get a new query embedding q:

   $$q_{new} = q_{old} + \sum_i p_i * v_i$$

5. The procedure above is called a "hop," in every iteration, we recurrently performs attention to get an internal embedding, and we also add residual connection from the original query embedding to the internal embedding to produce the new query embedding for next iteration. The weights for all embedding matrices are shared across hops.

The original purpose of End-To-End Memory Network is to summarize the whole passage by producing a embedding vector for the passage, since the attention mechanism instructs it to focus on specific sentences. In our implementation, since the HuggingFace API requires tokenized input sequence instead of continuous-valued embeddings, we need to slightly modify the way in which we employ the model.

Since End-To-End Memory Network produces a scores p_i for each individual sentence in each hop, we take the score at the last hop as the final output of the module. Then, we select the top k sentences that have the highest score. We concatenate and tokenize the selected sentences and input them into the generator model. During training, this extra sentence selector is trained end-to-end together with the generator.

One significant merit of this module is that the score assigned to each sentence is updated during the training; so as the training proceed, different sentences can be selected in different iteration, allowing the generator to see various contexts.

3.2 Dataset Descriptions

Given that our goal is to train a model that generates complex standardized test questions, we use question-answering datasets extracted from exams to train our models..

3.2.1 TOEFL-QA

The TOEFL dataset contains 963 passages and several multiple-choice questions for each passage. The questions are pulled from the Listening portion of the exam, and are in a multiple-choice format. The TOEFL questions are used as our goal for QG, our evaluation dataset. This is due to its complex questions and the prevalence of the TOEFL in the education industry.

3.2.2 RACE

RACE consists of more than 28,000 passages and 100,000 questions, which are collected from middle-school to high-school level English examinations in China. Each row contains an article, a multiple-choice question, four options, and the correct answer. This dataset is significantly larger than our TOEFL dataset.

3.3 Evaluation metrics

Evaluation can be performed both manually and automatically. Automatic methods include BLEU and BERTScore for the semantic similarities of generated questions and ground truth questions. Except for BLEU and BERTScore, we also made use of useful metrics such as METEOR and ROUGE.

BLEU[15], ROUGE[17] and METEOR[16] are similar metrics. To be more specific, BLEU is a metric which calculates score according to recall and precision using n-grams, while ROUGE depends on recall only. In this project, we used ROUGE1, meaning ROUGE with 1-grams. METEOR, an improvement on BLEU, modifies calculations with respect to recall and precision. Since all three metrics are used widely in fields of NLP, it is valuable to score output of the models with all three scores.

BERTScore is a special form of metric, because it takes in reference sentence and prediction sentence using embeddings, instead of matching word-by-word. Such embeddings is able to generate different vectors for the same word in different context, making score for prediction more flexible. After embedding, to calculate BERTScore, we combine the concept of cosine similarity and F-Score. Let x be the reference sentence, and $\hat{x}$ be the predicted sentence, by reducing cosine similarity, we have $R = \frac{1}{|x|} \sum_{x_i \in x} \max_{\hat{x}_j \in \hat{x}} x_i^T \hat{x}_j$ and $P = \frac{1}{|\hat{x}|} \sum_{\hat{x}_j \in \hat{x}} \max_{x_i \in x} x_i^T \hat{x}_j$, where R refers to maximized recall and P refers to maximizes precision. Then we use F1 scores to calculate BERTScore: $F = 2\frac{P \cdot R}{P+R}$. [9]

4 Experiment

Question generation is a particularly difficult field to methodically experiment and evaluate. In this report, we constrain our automatic evaluation methods to how well our model recreates a ground truth given nothing but a raw passage.

We fine-tune identical T5 models to compare, varying the input format and dataset. We train model variant 1 (model A) to use nothing but a context. Inputs to the model are limited to 512 tokens as defined by the pretrained model, but full TOEFL passages are much larger. To prevent truncating and losing information, during training, we take a ground truth answer and find the five most semantically similar sentences in the passage using BERTScore, and use that as the input context to our model. The other method of training is almost identical, but we also indicate the answer with a special token and prepend that to our inputs (model B). We also implemented another model that includes the learnable End-To-End Memory Network component, where the memory network automatically learns the relative importance of sentences given the reference question and select the five most relevant questions as the input context (model C).

In our experiments, we also compare varying training times on our relatively small TOEFL dataset. For the models trained on the RACE dataset, since it is significantly larger and thus more in-line with datasets in prominent literature, we simply fine tune for three epochs.

We use SGD with a fixed learning rate of 0.001 in all of our QG experiments. The only modifications to the publicly available T5-base model is the addition of two special tokens denoting answer and context respectively, following methodology from a different QG model on HuggingFace [28]. Our batch size is one due to the limit in computation resources.

During inference, we are not given questions or answers, and are only given the passage. For model A which only uses a context, we naively break the given passage into blocks of 3 sentences generate questions for each block to evaluate. For models B and C, we use the cause and effect model described above to find "effects" in blocks of sentences which are used as answers, then use the same blocks of sentences as context.

4.1 Generating TOEFL-like Questions

In the table below, we compare different inputs to our model. Given a passage, we compare two methods of generating questions. We generate many questions per passage, but there are only four to six ground truth questions. We compare each of these questions to each ground truth question and record the values. The values listed in the table below are the average of the top-5 most similar pairs. This measures how well our model can generalize to generate TOEFL-like questions and is an automatic indicator that other generated questions are likely also high-quality.

The notation in the table is as follows: B stands for base, referring to the baseline pretrained T5 model. TOEFL-trained models are indicated with a T. Race-trained models are indicated with an R. Models with answer are given an A, and models without do not. Epochs of training are denoted by the number after the dash. S denotes whether the model was trained with End-To-End Memory Network for sentences selection.

Model	BERTScore	BLEU	ROUGE	METEOR
T-B	66.9%	1.5%	13.1%	7.6%
T-10	69.0%	1.2%	15.5%	11.7%
T-50	76.0%	1.7%	39.4%	36.7%
T-100	81.4%	7.5%	49.4%	41.6%
TA-B	63.3%	1.4%	8.8%	5.7%
TA-10	67.9%	1.6%	17.3%	15.3%
TA-50	81.6%	3.2%	52.4%	48.0%
TA-100	79.9%	8.4%	44.5%	42.2%
TA-S100	81.3%	10.9%	53.2%	50.8%
R-B	82.4%	51.8%	80.0%	62.4%
R-3	91.7%	58.8%	95.2%	98.6%
RA-B	79.7%	25.3%	62.9%	56.8%
RA-3	94.4%	78.6%	100%	99.9%

4.2 Question Coherence and Human Evaluation

Automatic metrics have their limitations, which have been called into question in particular for Natural Language Generation tasks [21]. Many automatic metrics have difficulty recognizing different wordings or ways of stating text.[10]

Manual methods are also valuable, and question-answer pairs will be holistically judged by deep learning students on various criteria such as TOEFL-likeness and coherence.

Evaluators in this report are limited in this report to students that have previously studied and taken the TOEFL exam.

4.3 Answering the Generated Questions

For a complete pipeline, we need to answer the questions that are generated by our QG model. We use the pretrained t5-small that is further fine-tuned on the TOEFL-QA and RACE datasets to generate answers. The details of the fine-tuning process are described as follows.

Since questions generated by the QG model do not come with labels, we need to fine-tune our model on the original datasets. Although our question generation model focused more on coming up with TOEFL-style questions, it is limited to use only the TOEFL-QA dataset to train our QA model because of the small size of the dataset. Therefore, we also include the RACE dataset and mixed the training instances with those of the TOEFL-QA dataset. We fine-tuned our model on the mixed dataset for 2 epochs and gave it an additional epoch on the dataset that contains only TOEFL-QA instances. We used ADAM as the optimizer, with a learning rate of 0.0001. Our batch size is 4.

During the inference part, we feed our QA model with the questions that are generated by our QG model, alongside with the same passage as the context.

5 Results

5.1 Question Generation

From the chart above, we may see how our model did for TOEFL question generation as well as RACE. More specifically, for TOEFL-finetuned T5 model, after running 100 epochs, there is an exceptional increase in metric scores. It is able to reach BERTScore of about 81%, meaning that the model is producing answers very similar to the ground truth in terms of contexts. There is also an increase in BLEU score. BLEU is strict, since the score is corresponding to how similar the references are comparing to predictions, and thus it is normal to have a comparatively low BLEU score for TOEFL-fintuned T5 model, since the dataset is not large. Furthermore, the ROUGE score and METEOR score both have substantial increase, indicating an improving model.

We also performs ablation study on several components, like whether or not to include answer or selector in training, and whether or not to incorporate cause-and-effect model in inference. Except for models that only takes in contexts to train, we also have T5 model trained with answers using cause-and-effect model to extract the causations within contexts. During training, we find that there was a slight decrease in BERTScore, ROUGE and METEOR, while gained a significant increase in BLEU score. Since the reduction is slight comparing to how more BLEU score improved, the model is indeed improving. After trying such technique, we also tried End-to-End Memory Network Selector along with answers. In the end, the model performed better than with answers only. It reached a high BLEU score as well as ROUGE and METEOR, comparing to previous models.

Other than TOEFL, we also fine-tuned our model on RACE dataset, and it turns out the model works well for RACE. With merely 3 epochs, we are able to achieve BERTScore, ROUGE and METEOR over 0.9, and a BLEU score of over 0.5. The technique to train with answers also works well for RACE. We even achieved higher score for all metrics.

As mentioned in section 4.2, metrics have their limitations, and thus we also utilized human evaluation. Here is an example of a generated question using the TOEFL fine-tuned T5 model. An example is contracted from generated questions from *tpo-22-conversion-2*. The generated question is *What happened to the pianist when she was invited to join a jazz orchestra?*. By reading the context, we know that the pianist, referring to grandmother of the student, was out of conservatory when she was invited to join a jazz orchestra. This question has a low metric score since it is very different from the ground truth; however, the question is still grammatically correct, and it is answerable using the context. We have done this on some output of the question generation model, and find that the models are working well generating answerable questions.

In conclusion, our model works well for the TOEFL-trained model, in spite of the dataset's size, some progress is achieved. However, with the sizeable RACE dataset, we are able to achieve a high performance with our methodology.

5.2 Question Answering

Due to the lack of ground-truths for all good questions that could be asked of a passage, it is hard to evaluate the output of our question answering model based on some fixed metrics. Therefore, we used human evaluation to examine the quality of our machine-generated question and answer pairs. Some of the examples are presented as follows:

Example1: Passage Title: *tpo-22-conversation-1-1*
Auto-generated Question: *what is the professor's opinion?*
Response from the QA Model: *she is not sure whether the paper should be retracted.*

Example2: Passage Title: *tpo-25-lecture-1-6*
Auto-generated Question: *what does the professor imply about the translocation of animals?*
Response from the QA Model: *the translocation of animals is a cause of many changes in animal life.*

Example3: Passage Title: *tpo-23-lecture-2-14*
Auto-generated Question: *why does the professor mention that low thin clouds contribute to solar radiation?*
Response from the QA Model: *to illustrate the difference between albedos and ocean energy.*

The question generated from the context in Example1 is an instance of *conclusive* questions, which are frequently asked in standardized tests such as the TOEFL exam. Although there can be multiple valid answers to this kind of question, most options are wrong semantically in TOEFL, so the correct option is still clear. When encountering *conclusive* questions, our model is effective in terms of extracting relevant facts from the passage and constructing them as an answer. Although the auto-generated answers often differ from the ground-truth answers, from a human perspective, these answers can still be considered valid, given that they appeals to the important facts provided by the passage.

The second type of questions asks about *facts* in the article, where the second generated question can be a perfect example. Typically, among all kinds of questions, our model can handle this type of questions best, as illustrated by the response of the model in Example2.

Another type of questions focuses on *reasoning*. As it appears in Example3, *reasoning* questions often asks about the logical connections between statements made by the professor. The response from our QA model seems logical on the first look, but if one reads the passage more carefully, sometimes the response has little or no causal effect with the professor's claim. Although the QA model is able to choose relevant information from the passage, its ability of finding causal relationships is relatively limited. This is further discussed in section 6.2.

6 Discussion

6.1 Question Generation

All in all, the question generation models generate reasonable questions regarding the contexts with TOEFL-finetuned as well as RACE-finetuned models. As indicated in section 5.1, the score for both models are mostly increasing with some exceptions. These exceptions might be due to a lack of data. The TOEFL-QA dataset is small, and thus would cause issues such as overfit. However, even with limited dataset with TOEFL-QA, the model is able to generate common questions, such as *What is the lecture mainly about?*, which is a common question across TOEFL. The model is also able to generate questions with details, but is hard to generate ground truth, which is expected since the pattern of how TOEFL questions are framed is not strong, and the dataset is small.

However, by evaluating the questions generated with low metric scores manually, many of these are grammatically correct and answerable using the context. Thus, the model is performing better than it seems in metric scores, since question generation is not only limited to the ground truths, but also grammar and answerability.

6.2 Question Answering

In general, our question answering model gives decent answers for auto-generated questions, but there are several aspects to consider for making further improvements to the QA model. First, since its ability to identify causal relationships is limited, further researches could explore integrating the cause-effect module to the QA model as part of its prior knowledge. Being able to identify causal relationships in the context of the question can significantly enhance the model's ability to solve *reasoning* questions, as well as increasing its application value, as it would be tricky to identify flawed answers for *reasoning* questions otherwise.

Additionally, despite the lack of ground-truth answers for auto-generated questions, it is still possible to automatically evaluate the quality of the QA output. For example, using GANs[27] to classify ground-truth question-answer pairs against auto-generated pairs is worth-investigating, but these are beyond the scope of this paper.

7 Conclusion

The baseline model that we implement is the T5-base model open-sources on huggingface. We fine-tuned on the T5-base model with extra modules introduced such as the cause-and-effect extractor, BERTScore context selector and End-To-End Memory Network context selector. During inference, we only feed in the passage to the model, we use the cause and effect extractor described above to find "effects" in blocks of sentences which are used as answers, then use the same blocks of sentences as context to generate question.

For evaluation, we found that the incorporation of memory network selector does improve the result, meaning that the model is able to focus on the most relevant part in the input passage. Also, the model pretrained on RACE dataset performs better than TOEFL dataset, due to the fact that RACE is large, and the passages in RACE are relatively less complex compared to TOEFL. With human evaluation, we found that the question generator is able to generate some questions in TOEFL style, and the answer generator can generate something that requires reasoning rather rather than superficially copying some words in the original passage. This means that our work can be used to build question bank for similar standardized test, providing students and instructors with more preparation materials.

Currently, our model does not learn the statistical distribution for passage, question, and answer. For future improvements, we can incorporate module like VAE and GAN, such that the module can learn the underlying distribution behind data.

References

[1] Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems. 2017.

[2] Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

[3] Raffel, Colin, et al. "Exploring the limits of transfer learning with a unified text-to-text transformer." arXiv preprint arXiv:1910.10683 (2019).

[4] Raffel, Colin, et al. "Exploring the limits of transfer learning with a unified text-to-text transformer." arXiv preprint arXiv:1910.10683 (2019).

[5] Stasaski, Katherine, et al. "Automatically Generating Cause-and-Effect Questions from Passages." Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications. 2021.

[6] Kurdi, Ghader, et al. "A systematic review of automatic question generation for educational purposes." International Journal of Artificial Intelligence in Education 30.1 (2020): 121-204.

[7] Chen, Chia-Yin, Hsien-Chin Liou, and Jason S. Chang. "Fast–an automatic generation system for grammar tests." Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions. 2006.

[8] Rajpurkar, Pranav, et al. "Squad: 100,000+ questions for machine comprehension of text." arXiv preprint arXiv:1606.05250 (2016).

[9] Zhang, Tianyi, et al. "BERTScore: Evaluating Text Generation with BERT." arXiv preprint arxiv:1904.09675(2020)

[10] Nema, Preksha and Mitesh M. Khapra. "Towards a Better Metric for Evaluating Question Generation Systems." arXiv preprint arXiv:1808.10192 (2018)

[11] Shuailiang Zhang, et al. "DCMN+: Dual Co-Matching Network for Multi-choice Reading Comprehension", arXiv preprint arXiv:1908.11511 (2020)

[13] Sainbayar Sukhbaatar, et al. "End-to-End Memory Network", arXiv preprint arXiv:1503.08895 (2015)

[14] Patrick Lewis, et al. "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks", arXiv preprint arXiv:2005.11401 (2021)

[15] Kishore Papineni, et al. "BLEU: A Method for Automatic Evaluation of Machine Translation", Association for Computational Linguistics, Proceedings of the 40th Annual Meeting on Association for Computational Linguistics 2002

[16] Banerjee, Satanjeev and Alon Lavie. "METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments", Association for Computational Linguistics, Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization 2005.

[17] Lin, Chin-Yew. "ROUGE: A Package for Automatic Evaluation of Summaries", Association for Computational Linguistics, Text Summarization Branches Out 2004

[18] Lai, Guokun, et al. "RACE: Large-scale ReAding Comprehension Dataset From Examinations" arXiv preprint arXiv:1704.04683 (2017)

[19] Tseng, Bo-Hsiang, et al. "Towards machine comprehension of spoken content: Initial TOEFL listening comprehension test by machine" INTERSPEECH (2016)

[20] Chung, Yu-An, Hung-Yi Lee and James Glass. "Supervised and unsupervised transfer learning for question answering." NAACL HLT (2018)

[21] Belz, Anja, and Ehud Reiter. "Comparing automatic and human evaluation of NLG systems." 11th conference of the european chapter of the association for computational linguistics. 2006.

[22] Lindberg, David, et al. "Generating natural language questions to support learning on-line." Proceedings of the 14th European Workshop on Natural Language Generation. 2013.

[23] Mannem, Prashanth, Rashmi Prasad, and Aravind Joshi. "Question generation from paragraphs at UPenn: QGSTEC system description." Proceedings of QG2010: The Third Workshop on Question Generation. 2010.

[24] Jadallah. "Cause-Effect Detection for Software Requirements Based on Token Classification with BERT" Seminar Natural Language Processing for Software Engineering. https://huggingface.co/noahjadallah/cause-effect-detection 2021.

[25] Duan, Nan, et al. "Question generation for question answering." Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. 2017.

[26] Heilman, Michael, and Noah A. Smith. "Good question! statistical ranking for question generation." Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. 2010.

[27] Rao, Sudha, and Hal Daumé III. "Answer-based adversarial training for generating clarification questions." arXiv preprint arXiv:1904.02281 (2019).

[28] Montgomerie , Adam. “Amontgomerie/question_generator: An NLP System for Generating Reading Comprehension Questions.” GitHub, https://github.com/AMontgomerie/question_generator.

[29] Lan, Zhenzhong, et al. "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations.." Paper presented at the meeting of the ICLR, 2020.

[30] Sainbayar Sukhbaatar, et al. "End-To-End Memory Networks" arXiv preprint arXiv: 1503.08895 (2015).

[31] Lee, Dong Bok, et al. "Generating Diverse and Consistent QA pairs from Contexts with Information-Maximizing Hierarchical Conditional VAEs", Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020.

Appendix

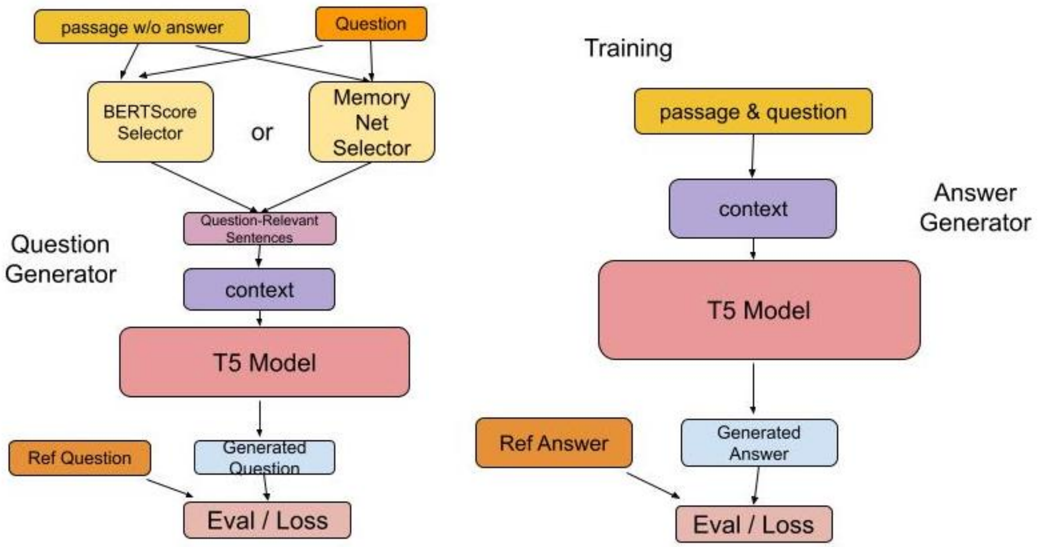

Figure 1: Training Pipeline: our model consists of a question generator and an answer generator; during training, the two components are trained separately. Note that the cause and effect generator is not included in training paradigm.

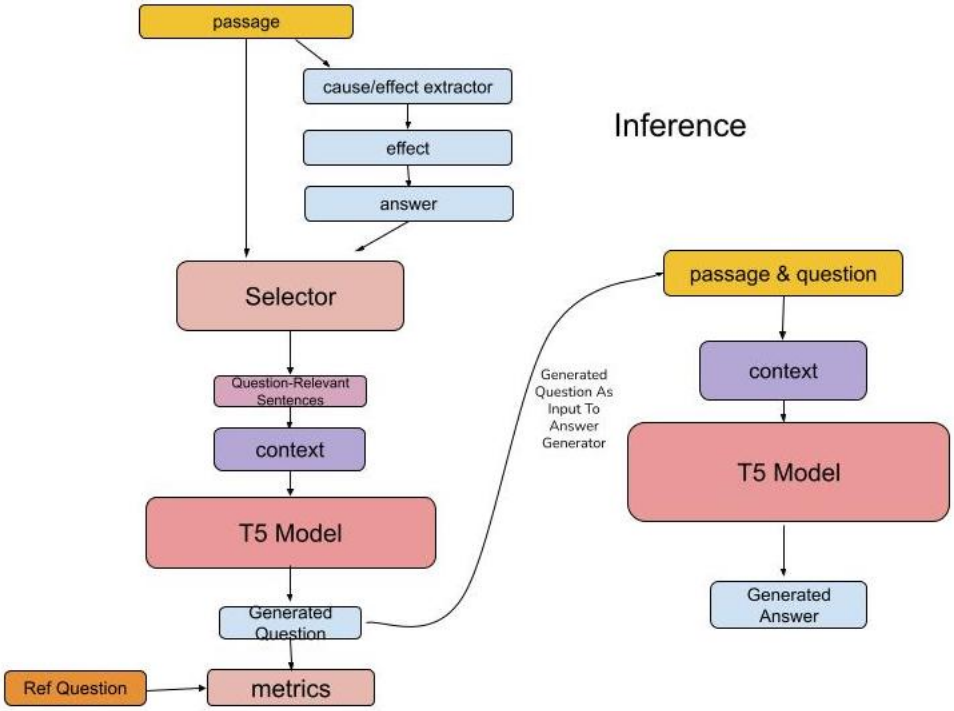

Figure 2: Inference Pipeline: during, the generated question is feeded as input to the answer generator. Since the input to the question generator is only the passage, the cause-and-effect extractor is employed to extract the effect in the passage, and use it as the answer for question generation.

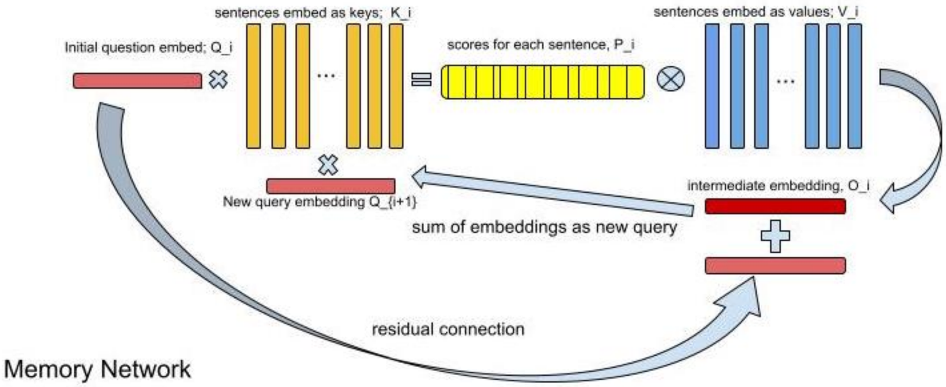
sentences embed as keys; K_i
Initial question embed; Q_i
scores for each sentence, P_i
sentences embed as values; V_i
New query embedding Q_{i+1}
intermediate embedding, O_i
sum of embeddings as new query
residual connection
Memory Network

Figure 3: End-To-End Memory Network produces a scores p_i for each individual sentence in each hop, we take the score at the last hop as the final output of the module. All the embedding weight matrices for key, value, and query are shared in RNN fashion.