Downstream Transformer Generation of Question-Answer Pairs with Preprocessing and Postprocessing Pipelines

Cheng Zhang and **Hao Zhang** and **Jie Wang**

Department. of Computer Science

University of Massachusetts, Lowell, MA 01854

{cheng_zhang, hao_zhang}@student.uml.edu, wang@cs.uml.edu

Abstract

We present a system called TP3 to perform a downstream task of transformers on generating question-answer pairs (QAPs) from a given article. TP3 first finetunes pretrained transformers on QAP datasets, then uses a preprocessing pipeline to select appropriate answers, feeds the relevant sentences and the answer to the finetuned transformer to generate candidate QAPs, and finally uses a postprocessing pipeline to filter inadequate QAPs. In particular, using pretrained T5 models as transformers and the SQuAD dataset as the finetruning dataset, we show that TP3 generates satisfactory number of QAPs with high qualities on the Gaokao-EN dataset.

1 Introduction

It is a grand challenge to generate adequate question-answer pairs (QAPs) from a given article. A QAP is adequate if both the question and the answer are contextually and grammatically correct and conform to native speakers, while the answer matches the question in the context of the article. Existing methods on question generation are based either on handcrafted features or on deep neural-network models. Methods of the former typically rely on grammar rules. However, no matter how many rules are formatted and used, there are always exceptions that these rules don't apply, resulting in generations of inadequate questions. Methods of the latter typically perform better, but may still generate incoherent questions. For instance, asking about clauses that express reasons or purposes may produce incoherent QAPs.

We present TP3 (Transformer with Preprocessing and Postprocessing Pipelines) for generating QAPs. It is a downstream task on a pretrained transformer that is finetuned on a QAP dataset, with a preprocessing pipeline to select appropriate answers and a postprocessing pipeline to filter undesirable questions. These pipelines are a combination of various NLP tools and algorithms. In particular, we finetune a pretrained T5-Large model (a Text-to-Text Transfer Transformer pretrained on a large dataset) (Raffel et al., 2020) on the SQuAD dataset (Rajpurkar et al., 2016) and show that it outperforms the state-of-the-art results under standard metrics.

For simplicity, we still use TP3 to denote this finetuned T5 transformer with the preprocessing and postprocessing pipelines. We show that it generates over 92% adequate QAPs among the 1,172 QAPs generated on the Gaokao-EN dataset of 85 articles of length ranging from 15 to 20 sentences each, collected from multiple Gaokao English tests for college entrance examinations.

The rest of the paper is organized as follows: We discuss in Section 2 related work on question generation. In Section 3, we describe how we finetune pretrained T5 models and in Section 4 we present our preprocessing and postprocessing pipelines. We provide in Section 5 evaluation results and conclude the paper in Section 6.

2 Related Works

Early methods on question generation are typically rule-based systems that transform a declarative sentence into a factual interrogative sentence (Heilman and Smith, 2009; Lindberg, 2013; Labutov et al., 2015). This approach includes methods to identify key phrases from input sentences, generate questions and answers using syntactic or semantic parsers and named entity analyzers, and transform declarative

sentences into interrogative sentences based on linguistic features and syntactic rules for different types of questions (Danon and Last, 2017; Khullar et al., 2018; Ali et al., 2010).

The quality of generated QAPs are evaluated either by human judges or by the following metrics: BLEU (Papineni et al., 2002a), ROUGE (Lin, 2004), and METEOR (Banerjee and Lavie, 2005), even though none of these metrics measure grammatical correctness of the questions being generated.

Recent advances of neural-network research provide new tools to build generative models. For example, the attention mechanism can help determine what content in a sentence should be asked (Luong et al., 2015), and the sequence-to-sequence (Bahdanau et al., 2014; Cho et al., 2014) and the long short-term memory (Sak et al., 2014) mechanisms are used to generate words to form a question (see, e.g., (Du et al., 2017; Duan et al., 2017; Harrison and Walker, 2018; Sachan and Xing, 2018)). These models generate questions without the corresponding correct answers. To address this issue, researchers have explored ways to encode a passage (a sentence or multiple sentences) and an answer word (or a phrase) as input, and determine what questions are to be generated for a given answer (Zhou et al., 2018; Zhao et al., 2018; Song et al., 2018). However, as pointed out by Kim et al. (Kim et al., 2019), these methods could generate answer-revealing questions, namely, questions contain in them the corresponding answers. They then devised a new method by encoding answers separately, at the expense of learning substantially more parameters.

More recently, researchers have explored how to use pretrained transformers to generate answer-aware questions (Dong et al., 2019; Zhang and Bansal, 2019; Zhou et al., 2019; Qi et al., 2020b; Su et al., 2020; Lelkes et al., 2021). For example, Kettip et al. (Kriangchaivech and Wangperawong, 2019) presented an architecture for a transformer to generate questions. Rather than fully encoding the context and answers as they appear in the dataset, they applied certain transformations such as the change of named entities both on the context and the answer. Lopez et al. (Lopez et al., 2020) finetuned the pretrained GPT-2 (Radford et al., 2019) transformer without using any additional complex components or features to enhance its performance. Chen (Chen et al., 2020) described a fully transformer-based reinforcement learning generator evaluator architecture to generate questions.

The recent introduction of T5 (Raffel et al., 2020) has escalated NLP research in a number of ways. T5 is a encoder-decoder text-to-text transformer using the teacher forcing method on a wide variety of NLP tasks, including text classification, question answering, machine translation, and abstractive summarization. Unlike other transformer models (e.g. GPT-2 (Radford et al., 2019)) that take in text data after converting them to corresponding numerical embeddings, T5 handles each task by taking in data in the form of text and producing text outputs.

Taking the advantage of pretrained T5, Lidiya et al. (Murakhovs'ka et al., 2021a) combined nine question-answering datasets to finetune a single T5 model and evaluated generated questions using a new semantic measure called BERTScore (Zhang et al., 2020). Their method achieves so far the best results. We present a finetuned T5 model on a single SQuAD dataset to produce better results.

3   Description of TP3

We describe how we train and finetune a pretrained T5 transformer for our downstream task of question generation and use a combination of various NLP tools and algorithms to build the preprocessing and postprocessing pipelines for generating QAPs.

There are a number of public QAP datasets available for fine-tuning T5, including RACE (Lai et al., 2017), CoQA (Reddy et al., 2019), and SQuAD (Rajpurkar et al., 2016). RACE is a large-scale dataset collected from Gaokao English examinations over the years, where Gaokao is the national college entrance examinations held once every year in mainland China. It consists of more than 28,000 passages and nearly 100,000 questions, including cloze questions. CoQA is a conversational-style question-answer dataset. It contains a series of interconnected questions and answers in conversations. SQuAD is a reading comprehension dataset, consisting of more than 100,000 QAPs posted by crowdworkers on a set of Wikipedia articles.

Among these datasets, SQuAD is more commonly used in the question generation research. We use SQuAD to finetune pretrained T5 models. For each QAP and the corresponding context extracted from

the SQuAD training dataset, we concatenate the answer and the context with markings in the format of $\langle answer \rangle answer\_text \langle context \rangle context\_text$ as input, with the question as the target, where the context is the entire article for the QAP in SQuAD. We then set the maximum input length to 512 and the target length to 128 to avoid infinite loops and repetitions of target outputs. We feed the concatenated text input and question target into a pretrained T5 model for fine-tuning and use AdamW (Loshchilov and Hutter, 2019) as an optimizer with various learning rates to obtain a better model.

To explore various learning rates, we first use automatic evaluation methods to narrow down a smaller range of the learning rates and then use human judges to determine the best learning rate. In particular, we first finetune the base model with a learning rate of 1.905×10^{-3} and the large model with a learning rate of 4.365×10^{-4}. The learning rates are calculated using the Cyclical Learning Rates (CLR) method (Smith, 2017), which is used to find automatically the best global learning rate. Evaluated by human judges, we found that the best learning rate calculated by CLR is always larger than the actual best learning rate in our experiments.

We then finetune T5-Base and T5-Large with dynamic learning rates from the learning rate calculated by CLR with a reduced learning rate for each epoch. For example, we finetune T5-Base starting from a learning rate of 1.905×10^{-3} and multiply the previous learning rate by 0.5 for the current epoch until the learning rate of 1.86×10^{-6} is reached. Likewise, we finetune T5-Large in the same way starting from 4.365×10^{-4} until the learning rate of 1.364×10^{-5} is reached. However, the generated results are still below expectations.

We therefore proceed to finetune the models with various learning rates we choose. In particular, we first finetune T5-Base with a learning rate from 10^{-3} to 10^{-4} with a 2.5×10^{-4} decrement for each epoch, and from 10^{-4} to 10^{-5} with a 2.5×10^{-5} decrement for each epoch. Likewise, we finetune T5-Large with a learning rate from 10^{-4} to 10^{-5} with a 2×10^{-5} decrement for each epoch, and from 10^{-5} to 10^{-6} with a 2×10^{-6} decrement for each epoch.

Evaluated using BLEU (Papineni et al., 2002b), ROUGE (Lin, 2004), METEOR (Banerjee and Lavie, 2005) and BERTScore (Zhang et al., 2020), we find that the learning rates ranging from 10^{-4} to 10^{-5} for T5-Base and the learning rates ranging from 10^{-5} to 10^{-6} for T5-Large perform better. Moreover, as expected, the overall performance of T5-Large is better than T5-Base.

Tables 1 and 2 depict the measurement results for T5-Base and T5-Large, respectively, where R1, R2, RL, and RLsum stand for, respectively, Rouge-1, Rouge-2, Rouge-L, and Rouge-Lsum. The boldfaced number indicates the best in its column. It is evident that T5-Base with the learning rate of 3×10^{-5} and T5-Large with the learning rate of 8×10^{-6} produce the best results. For convenience, we refer to these two finetuned models as T5-Base-SQuAD$_1$ and T5-Large-SQuAD$_1$ to distinguish them with the existing T5-Base-SQuAD model. We sometimes also denote T5-Base-SQuAD$_1$ as T5-SQUAD$_1$ when there is no confusion of what size of the dataset is used to pretrain T5.

Table 1: Automatic Evaluation of T5-Base-SQuAD$_1$

Learning Rate	BLEU	R1	R2	RL	RLsum	METEOR	BERTScore	Average
5e-5	20.01	50.71	28.38	46.59	46.61	45.46	51.51	41.32
3e-5	**22.63**	**54.90**	**32.22**	**50.97**	**50.99**	**48.98**	**55.82**	**45.22**
2.5e-5	22.50	54.36	31.93	50.49	50.50	48.64	55.61	44.86
1e-5	20.17	50.46	28.38	46.79	46.81	44.97	51.82	41.34
Dynamic	20.57	51.88	28.99	47.67	47.68	47.38	53.34	42.50

4 Processing Pipelines

The processing pipelines consist of preprocessing to select appropriate answers, question generation, and postprocessing to filter undesirable questions (see Fig. 1).

Table 2: Automatic Evaluation on T5-Large-SQuAD$_1$

Learning Rate	BLEU	R1	R2	RL	RLsum	METEOR	BERTScore	Average
3e-5	23.01	54.49	31.92	50.51	50.51	50.00	56.19	45.23
1e-5	23.66	51.88	32.88	51.43	51.42	50.53	56.65	45.50
8e-6	23.83	**55.48**	**33.08**	**51.58**	**51.58**	50.61	**56.94**	**46.15**
6e-6	**23.84**	55.24	32.91	51.35	51.35	**50.70**	56.57	45.99
Dynamic	20.86	52.00	29.46	48.03	48.03	47.68	53.85	42.84

Figure 1: TP3 Architecture

4.1 Preprocessing

We observe that how to choose an answer would affect the quality of a question generated for the answer. We use a combination of NLP tools and algorithms to construct a preprocessing pipeline for selecting appropriate answers as follows:

1. *Remove unsuitable sentences.* We first remove all interrogative and imperative sentences from the given article. We may do so by, for instance, simply removing any sentence that begins with a WH word or a verb and any sentence that ends with a question mark. We then use semantic-role labeling (Shi and Lin, 2019) to analyze sentences and remove those that do not have any one of the following semantic-role tags: subject, verb, and object. For each remaining sentence, if the total number of words contained in it, excluding stop words, is less than 4, then remove this sentence. We then label the remaining sentences as *suitable* sentences.

2. *Remove candidate answers with inappropriate semantic-role labels.* Nouns and phrasal nouns are candidate answers. But not any noun or phrasal noun would be suitable to be an answer. We'd want a candidate answer to associate with a specific meaning. Specifically, if a noun in a suitable sentence is identified as a name entity (Peters et al., 2017) or has a semantic-role label in the set of {ARG, TMP, LOC, MNR, CAU, DIR}, then keep it as a candidate answer and remove the rest, where ARG represents subject or object, TMP represents time, LOC represents location, MNR represents manner, CAU represents cause, and DIR represents direction. If a few candidate nouns occur consecutively, we treat the sequence of these nouns as a candidate answer phrase.

   For example, in the sentence "The engineers at the Massachusetts Institute of Technology (MIT) have taken it a step further changing the actual composition of plants in order to get them to perform diverse, even unusual functions", the phrase "Massachusetts Institute of Technology" is recognized as a named entity, without a semantic-role label. Thus, it should not be selected as an answer. If it is selected, then the following QAP ("Where is MIT located", "Massachusetts Institute of Technology") will be generated, which is inadequate.

3. *Remove or prune answers with inadequate POS tags.* Using semantic-role labels to identify what nouns to keep does not always work. For example, the phrasal noun "This widget" in the sentence "This widget is more technologically advanced now" has a semantic-role label of ARG1 (subject), which leads to the generation of the following question: "What widget is more technologically advanced now?" It is evident that this QAP is inadequate even though it is grammatically correct. Note that "This" has a POS (part-of-speech) tag of PDT (predeterminer). For another example, while the word "now" in the sentence has a semantic-role label of TMP (time), its POS tag is RB (adverb). In general, we remove nouns with a POS tag in {RB, RP, CC, DT, IN, MD, PDT, PRP, WP, WDT, WRB} or prune words with such a POS tag at either end of a phrasal noun. After this

treatment, the candidate answer "now" is removed and the candidate answer phrase "This widget" is pruned to "widget". For this answer and the input sentence, the following question is generated: "What is more technologically advanced now?" Evidently this question is more adequate.

4. *Remove common answers.* We observe that certain candidate answers, such as "anyone", "people", and "stuff", would often lead to generation of inadequate questions. Such words tend to be common words that should be removed. We do so by looking up the probabilities of 1-grams from the Google Books Ngram Dataset (Michel et al., 2011). If the probability of a noun word is greater than 0.15%, we remove its candidacy. Likewise, we may also treat noun phrases by looking up the probabilities n-grams for $n > 1$, but doing so would incur much more processing time.

5. *Filtering answers appearing in clauses.* We observe that a candidate answer appearing in the latter part of a clause would often lead to a generation of an inadequate QAP. Such candidate answers would appear at lower levels in a dependency tree. We use the following procedure to identify such candidate answers: For each remaining sentence s, we first generate its dependency tree (Varga and Ha, 2010). Let h_s be the height of the tree. Suppose that a candidate answer a appears in a clause contained in s. If a is a single noun, let its height in the tree be h_a. If a is a phrasal noun, let the average height of the heights of the words contained in a be h_a. If $h_a \geq \frac{2}{3} h_s$, then remove a.

   Take the following sentence as an example: "While I tend to buy a lot of books, these three were given to me as gifts, which might add to the meaning I attach to them." In this sentences, the following noun "gifts" and phrasal nouns "a lot of books" and "the meaning I attach to them"are labeled as object. However, T5 resolves multiple objects poorly, and if we choose "the meaning I attach to them" as an answer, T5 will generate the following question: "What did the gifts add to the books", which is inadequate. Since this phrasal noun appears in a clause and at a lower level of the dependency tree, it is removed from being selected as a candidate answer.

6. *Removing redundant answers.* If a candidate answer word or phrase is contained in another candidate answer phrase and appear in the same sentence, we extract from the dependency tree of the sentence the subtree T_s for the shorter candidate phrase and subtree T_l for the longer candidate phrase, then T_s is also a subtree of T_l. If T_s and T_l share the same root, then the shorter candidate answer is more syntactically important than the longer one, and so we remove the longer candidate answer. Otherwise, remove the shorter candidate answer.

   Take the sentence "The longest track and field event at the Summer Olympics is the 50-kilometer race walk, which is about five miles longer than the marathon" as an example. The shorter phrase "Summer Olympics" is recognized as a named entity, which leads to the generation of the following inadequate QAP: ("What is the longest track and field event", "Summer Olympics). On the other hand, the longer phrase "The longest track and field event at the Summer Olympics" is labeled as subject for its semantic role, which leads to the generation of the following adequate QAP: ("What is the 50-kilometer race walk", "The longest track and field event at the Summer Olympics"). Since the root word for the longer phrase is "event" that is not contained in the shorter phrase, so the shorter phrase is removed to avoid generating the inadequate QAP.

4.2 Question generation

After extracting all candidate answers from the preprocessing pipeline, for each answer extracted, we use three adjacent sentences as the context, with the middle sentence containing the answer, and concatenate the answer and the context with marks into the following format as input to a fine-turned T5 model: $\langle answer \rangle$answer_text$\langle context \rangle$context_text, to generate candidate questions. We note that the greedy search in the decoder of the T5 model does not guarantee the optimal result, we use beam search with 3 beams to select the word sequences with the top 3 probabilities from the probability distribution and acquire 3 candidate questions. We then concatenate each candidate question with the corresponding answer as a new sentence and generate an embedding vector representation for it using the pretrained RoBERTa-Large model (Liu et al., 2019; Reimers and Gurevych, 2019), and select the most semantically similar question to the context as the final target question.

4.3 Postprocessing

Recall that in the preprocessing pipeline, we have removed inappropriate candidate answers. However, some of the remaining answers may still lead to generating inappropriate questions. Thus, in the post-processing pipeline, we proceed to remove inadequate questions as follows:

1. *Remove questions that contain the answers.* Remove a question if the corresponding answer or the main body of the answer is contained in the question. If the answer includes a clause, we extract the main body of the answer as follows: Parse the answer to constituency tree (Joshi et al., 2018) and remove the subtree rooted with a subordinate clause label SBAR, the remaining part of the phrase is the main body of the answer.

   For example, in the sentence "The first, which I take to reading every spring is Ernest Hemningway's A Moveable Feast", "The first, which I take to reading every spring" is labeled as subject. Using it as a candidate answer generates an inadequate question for the answer "What is the first book I reread?" Note that the phrase "The first" can be extracted as the main body of the answer, which is contained in the question. Thus, this QAP is removed.

2. *Remove short questions.* If the generated question, after removing stop words, consists of only one word, then remove the question. For example, "What is it?" and "Who is she?" will be removed because after removing stop words, the former becomes "What" and the latter becomes "Who". On the other hand, "Where is Boston?" will remain.

3. *Remove unsuitable questions.* Recall that we generate the question from the adjacent three sentences in the article, with the middle sentence containing the answer. However, the middle sentence may not be the only sentence containing the answer. In other words, the first or the last sentences may also contain the answer. Assuming that all three sentences contain the answer, our finetuned T5 transformer may generate a question based on the first sentence or the last sentence. If the first sentence or the last sentence is not a suitable sentence we labeled in the preprocessing pipeline, the question being generated may be in appropriate. We'd want to make sure that the question is generated for a suitable sentence. For this purpose, we first identify which sentence the question is generated for. In particular, let s_i for $i = 1, 2, 3$ be the 3 sentences and (q, a) be the question generated for answer a. Let QA denote the union of the set of words in q and the set of words in a. Likewise, let S_i be the set of words in s_i. If $QA \cap S_i$ is the largest among the other two intersections, then q is likely generated from s_i for a. If s_i is not suitable, then remove q.

   Note that we may also consider word sequences in addition to word sets. For example, we may consider longest common subsequences or longest common substrings when comparing two word sequences. But in our experiments, they don't seem to add extra benefits.

5 Evaluations

To evaluate the quality of QAPs generated by TP3-Base and TP3-Large, we use the standard automatic evaluation metrics as well as human judgments.

5.1 Automatic evaluations

We first compare T5-SQuAD$_1$ with the exiting QG models with the standard automatic evaluation metrics as before: BLEU, ROUGE-1 (R1), ROUGE-2 (R2), ROUGE-L (RL), ROUGE-LSum (RLsum), METEOR (MTR), and BERTScore (BScore). Since most existing QG models are based on pretrained transformers with the base dataset, we will compare T5-Base-SQuAD$_1$ with the existing QG models.

Table 3 shows automatic evaluation comparison results with ProphetNet (Qi et al., 2020a), BART (Lewis et al., 2020), T5 (Raffel et al., 2020) and MixQG (Murakhovs'ka et al., 2021b). BART-SQuAD, T5-SQuAD, and MixQG-SQuAD are corresponding models finetuned on the SQuAD dataset. BART-hl and T5-hl are augmented models using the "highlight" encoding scheme introduced by Chan and Fan (Chan and Fan, 2019).

Table 3: Automatic evaluation results

Model	Size	BLEU	R1	R2	RL	RLsum	MTR	BScore	Average
ProphetNet	Large	22.88	51.37	29.48	47.11	47.09	41.46	49.31	41.24
BART-hl	Base	21.13	51.88	29.43	48.00	48.01	40.23	54.33	41.86
BART-SQuAD	Base	22.09	52.75	30.56	48.79	48.78	41.39	54.86	42.75
T5-hl	Base	23.19	53.52	31.22	49.40	49.40	42.68	55.48	43.56
T5-SQuAD	Base	**23.74**	54.12	31.84	49.82	49.81	43.63	55.68	44.09
MixQG$_1$	Base	23.53	54.39	32.06	50.05	50.02	43.83	55.66	44.22
MixQG$_2$	Base	23.74	54.28	32.23	50.35	50.34	43.91	55.71	44.37
MixQG-SQuAD	Base	23.46	54.48	32.18	50.14	50.10	44.15	**55.82**	44.33
T5-SQuAD$_1$	Base	22.62	**54.87**	**32.20**	**50.99**	**50.98**	**48.98**	**55.82**	**45.21**

The results of MixQG$_1$ were presented in the original paper (Murakhovs'ka et al., 2021b), and the results of MixQG$_2$ were computed by us using the pretrained model posted on HuggingFace (https://huggingface.co/Salesforce/mixqg-base). The results show that, except BLEU, T5-SQuAD$_1$ outperforms all other models on the ROUGE and METEOR metrics, produces the same BERTScore score as that of MixQG-SQuAD. Overall, T5-SQuAD$_1$ performs better than all the models in comparison.

5.2 Manual evaluations of TP3

A number of publications (e.g., see (Callison-Burch et al., 2006; Liu et al., 2016; Nema and Khapra, 2018)) have shown that the aforementioned automatic evaluation metrics based on n-gram similarities do not always correlate well with human judgments about the answerability of a question. Thus, we'd also need to use human experts to evaluate the qualities of QAPs generated by TP3. We do so on the Gaokao-EN dataset consisting of 85 articles, where each article contains 15 to 20 sentences. We chose Gaokao-EN because expert evaluations are provided to us from a project we work on. Table 4 depicts the evaluation results. Title abbreviations are explained below, where the numbers in boldface are the best in the corresponding columns:

1. **Total** means the total number of QAPs generated by TP3.

2. **ADQT** means the total number of adequate QAPs. These QAPs can be directly used without any modification.

3. **ACPT** means the total number of QAPs where the question, while semantically correct, contains a minor English issue that can be corrected with a minor effort. For example, a question may simply be missing a word or a phrase at the end. Such QAPs may be deemed acceptable.

4. **UA** means unacceptable QAPs.

5. **ADQT-R** means the ratio of the adequate QAPs over all the QAPs being generated.

6. **ACPT-R** means the ratio of the adequate and acceptable QAPs over all the QAPs being generated.

Table 4: Manual evaluation results for TP3-Base and TP3-Large over Gaokao-EN

TP3	Learning Rate	Total	ADQT	ACPT	UA	ADQT-R	ACPT-R
Base	3e-5	**1290**	1145	**63**	82	88.76	93.64
Large	3e-5	1287	1162	49	76	90.29	94.09
	1e-5	1271	1166	39	76	91.74	94.81
	8e-6	1270	1162	39	69	91.50	94.57
	6e-6	1273	**1172**	45	**56**	**92.07**	**95.60**
	Dynamic	1288	1116	51	121	86.65	90.61

6 Conclusions

We presented a downstream task of transformers on generating question-answer pairs by finetuning pretrained T5 models with preprocessing and postprocess pipelines, and generate a satisfactory number of adequate QAPs for a given article with high qualities. To facilitate reproduction and further investigation, we have released the source code at https://github.com/zhangchengx/T5-Fine-Tuning-for-Question-Generation and the model at https://huggingface.co/ZhangCheng/T5-Base-finetuned-for-Question-Generation. The Gaokao-EN dataset and the human judgments of QAPs are also available at https://github.com/zhangchengx/Gaokao-EN

With an improved transformer it is possible to improve both the number and qualities of QAPs being generated. It's also possible to strengthen the preprocessing and postprocessing pipelines. For example, in addition to using a 1-gram language model to determine if a candidate answer would be appropriate, we may develop a more efficient method to use n-gram language models for checking a candidate answer being a phrasal noun. Also, when we feed a context to a transformer, in addition to feeding the model with three consecutive sentences in the article as we currently do, there are other ways to select sentences. For example, we may consider clustering similar sentences and rank them three at a time, such that the sentence in the middle contains the selected candidate answer. Another direction would be to explore how to generate QAPs for candidate answers that appear at lower levels of dependency trees. These issues deserve further investigations.

References

Ali, H., Chali, Y., and Hasan, S. A. (2010). Automation of question generation from sentences. In *Proceedings of QG2010: The Third Workshop on Question Generation*, pages 58–67.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Banerjee, S. and Lavie, A. (2005). METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Callison-Burch, C., Osborne, M., and Koehn, P. (2006). Re-evaluating the role of Bleu in machine translation research. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 249–256, Trento, Italy. Association for Computational Linguistics.

Chan, Y.-H. and Fan, Y.-C. (2019). A recurrent BERT-based model for question generation. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 154–162.

Chen, Y., Wu, L., and Zaki, M. J. (2020). Reinforcement learning based graph-to-sequence model for natural question generation. *ArXiv*, abs/1908.04942.

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Danon, G. and Last, M. (2017). A syntactic approach to domain-specific automatic question generation. *arXiv preprint arXiv:1712.09827*.

Dong, L., Yang, N., Wang, W., Wei, F., Liu, X., Wang, Y., Gao, J., Zhou, M., and Hon, H.-W. (2019). Unified language model pre-training for natural language understanding and generation. In *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*.

Du, X., Shao, J., and Cardie, C. (2017). Learning to ask: neural question generation for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1342–1352, Vancouver, Canada. Association for Computational Linguistics.

Duan, N., Tang, D., Chen, P., and Zhou, M. (2017). Question generation for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 866–874, Copenhagen, Denmark. Association for Computational Linguistics.

Harrison, V. and Walker, M. (2018). Neural generation of diverse questions using answer focus, contextual and linguistic features. In *Proceedings of the 11th International Conference on Natural Language Generation*. Association for Computational Linguistics.

Heilman, M. and Smith, N. A. (2009). Question generation via overgenerating transformations and ranking. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA LANGUAGE TECHNOLOGIES INST.

Joshi, V., Peters, M. E., and Hopkins, M. (2018). Extending a parser to distant domains using a few dozen partially annotated examples. In *ACL*.

Khullar, P., Rachna, K., Hase, M., and Shrivastava, M. (2018). Automatic question generation using relative pronouns and adverbs. In *Proceedings of ACL 2018, Student Research Workshop*, pages 153–158.

Kim, Y., Lee, H., Shin, J., and Jung, K. (2019). Improving neural question generation using answer separation. In *Association for the Advancement of Artificial Intelligence (AAAI)*, volume 33, page 6602–6609. Association for the Advancement of Artificial Intelligence (AAAI).

Kriangchaivech, K. and Wangperawong, A. (2019). Question generation by transformers.

Labutov, I., Basu, S., and Vanderwende, L. (2015). Deep questions without deep understanding. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 889–898, Beijing, China. Association for Computational Linguistics.

Lai, G., Xie, Q., Liu, H., Yang, Y., and Hovy, E. (2017). Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*.

Lelkes, A. D., Tran, V. Q., and Yu, C. (2021). Quiz-style question generation for news stories. In *Proceedings of the Web Conference 2021*, WWW '21, page 2501–2511, New York, NY, USA. Association for Computing Machinery.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Lin, C.-Y. (2004). ROUGE: a package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Lindberg, D. L. (2013). *Automatic question generation from text for self-directed learning*. PhD thesis, Applied Sciences: School of Computing Science.

Liu, C.-W., Lowe, R., Serban, I., Noseworthy, M., Charlin, L., and Pineau, J. (2016). How NOT to evaluate your dialogue system: an empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2122–2132, Austin, Texas. Association for Computational Linguistics.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Lopez, L. E., Cruz, D. K., Cruz, J. C. B., and Cheng, C. K. (2020). Transformer-based end-to-end question generation. *ArXiv*, abs/2005.01107.

Loshchilov, I. and Hutter, F. (2019). Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Luong, T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Michel, J.-B., Shen, Y. K., Aiden, A. P., Veres, A., Gray, M. K., null null, Pickett, J. P., Hoiberg, D., Clancy, D., Norvig, P., Orwant, J., Pinker, S., Nowak, M. A., and Aiden, E. L. (2011). Quantitative analysis of culture using millions of digitized books. *Science*, 331(6014):176–182.

Murakhovs'ka, L., Wu, C.-S., Niu, T., Liu, W., and Xiong, C. (2021a). Mixqg: Neural question generation with mixed answer types.

Murakhovs'ka, L., Wu, C.-S., Niu, T., Liu, W., and Xiong, C. (2021b). Mixqg: Neural question generation with mixed answer types.

Nema, P. and Khapra, M. M. (2018). Towards a better metric for evaluating question generation systems. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3950–3959, Brussels, Belgium. Association for Computational Linguistics.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002a). BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL'02, page 311–318, USA. Association for Computational Linguistics.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002b). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Peters, M. E., Ammar, W., Bhagavatula, C., and Power, R. (2017). Semi-supervised sequence tagging with bidirectional language models. In *ACL*.

Qi, W., Yan, Y., Gong, Y., Liu, D., Duan, N., Chen, J., Zhang, R., and Zhou, M. (2020a). Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 2401–2410.

Qi, W., Yan, Y., Gong, Y., Liu, D., Duan, N., Chen, J., Zhang, R., and Zhou, M. (2020b). ProphetNet: Predicting future n-gram for sequence-to-SequencePre-training. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2401–2410, Online. Association for Computational Linguistics.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392. Association for Computational Linguistics.

Reddy, S., Chen, D., and Manning, C. D. (2019). Coqa: A conversational question answering challenge.

Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Sachan, M. and Xing, E. (2018). Self-training for jointly learning to ask and nnswer questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 629–640, New Orleans, Louisiana. Association for Computational Linguistics.

Sak, H., Senior, A. W., and Beaufays, F. (2014). Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *Proccedings of the 15th Annual Conference of the International Speech Communication Association*.

Shi, P. and Lin, J. J. (2019). Simple bert models for relation extraction and semantic role labeling. *ArXiv*, abs/1904.05255.

Smith, L. N. (2017). Cyclical learning rates for training neural networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 464–472.

Song, L., Wang, Z., Hamza, W., Zhang, Y., and Gildea, D. (2018). Leveraging context information for natural question generation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 569–574, New Orleans, Louisiana. Association for Computational Linguistics.

Su, D., Xu, Y., Dai, W., Ji, Z., Yu, T., and Fung, P. (2020). Multi-hop question generation with graph convolutional network. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4636–4647, Online. Association for Computational Linguistics.

Varga, A. and Ha, L. A. (2010). Wlv: a question generation system for the QGSTEC 2010 task b. *Boyer & Piwek (2010)*, pages 80–83.

Zhang, S. and Bansal, M. (2019). Addressing semantic drift in question generation for semi-supervised question answering. In *EMNLP*.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. (2020). Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Zhao, Y., Ni, X., Ding, Y., and Ke, Q. (2018). Paragraph-level neural question generation with maxout pointer and gated self-attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3901–3910, Brussels, Belgium. Association for Computational Linguistics.

Zhou, Q., Yang, N., Wei, F., Tan, C., Bao, H., and Zhou, M. (2018). Neural question generation from text: a preliminary study. In Huang, X., Jiang, J., Zhao, D., Feng, Y., and Hong, Y., editors, *Natural Language Processing and Chinese Computing*, pages 662–671, Cham. Springer International Publishing.

Zhou, W., Zhang, M., and Wu, Y. (2019). Question-type driven question generation. In *EMNLP*.