EasyChair Preprint

№ 8739

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

Cheng Zhang, Hao Zhang, Yicheng Sun and Jie Wang

August 29, 2022

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

Cheng Zhang
Hao Zhang
Yicheng Sun
Jie Wang
{cheng_zhang,hao_zhang,yicheng_sun}@student.uml.edu
wang@cs.uml.edu
University of Massachusetts
Lowell, MA, USA

ABSTRACT

We present a method to perform a downstream task of transformers on generating question-answer pairs (QAPs) from a given article. We first finetune pretrained transformers on QAP datasets. We then use a preprocessing pipeline to select appropriate answers from the article, and feed each answer and the relevant context to the finetuned transformer to generate a candidate QAP. Finally we use a postprocessing pipeline to filter inadequate QAPs. In particular, using pretrained T5 models as transformers and the SQuAD dataset as the finetruning dataset, we obtain a finetuned T5 model that outperforms previous models on standard performance measures over the SQuAD dataset. We then show that our method based on this finetuned model generates a satisfactory number of QAPs with high qualities on the Gaokao-EN dataset assessed by human judges.

CCS CONCEPTS

• **Computing methodologies** → **Natural language processing**.

KEYWORDS

question generation, neural networks, natural language processing, natural language generation, information extraction

1 INTRODUCTION

Generating adequate question-answer pairs (QAPs) from a given article is a challenging task. A QAP is adequate if both the question and the answer are contextually and grammatically correct, conform to native speakers, and 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-net models. Methods of the former typically rely on grammar rules. However, no matter how many rules are formatted, there are always exceptions that these rules don't apply, leading to inadequate QAPs. Methods of the latter typically perform better, but may still generate inadequate QAPs. For instance, asking about clauses that express reasons or purposes may produce inadequate QAPs.

We present a method called TP3 (Transformer with Preprocessing and Postprocessing Pipelines) for generating QAPs in English. 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 (a Text-to-Text Transfer Transformer pretrained on a large dataset) [39] model on the SQuAD dataset [40] and show that it outperforms the state-of-the-art results under standard metrics.

We refer to TP3 based on this finetuned T5 model as T5P3. We apply T5P3 to an unseen dataset called Gaokao-EN, which consists of 75 articles of length between 15 and 20 sentences in each article, collected from multiple Gaokao English tests for college entrance examinations. We show that, assessed by human judges, T5P3 generates, on the Gaokao-EN dataset, over 83% adequate QAPs and over 92% acceptable (QAPs) among the 1,271 QAPs generated, where acceptable QAPs include QAPs that are adequate and almost adequate. A QAP is almost adequate if everything else is adequate except that the question contains a minor language issue that can be easily fixed with a minimum effort to become adequate. Such QAPs are used to help assess students' reading comprehension skills.

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 for T5P3. 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 factual declarative sentence into an interrogative sentence [13, 18, 23], including 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 [1, 8, 15].

The quality of generated QAPs are evaluated either by human judges without labeled datasets as references, or by the following standard metrics against some references: BLEU [33], ROUGE [22], and METEOR [3], even though none of these metrics measure grammatical correctness of the questions being generated.

Recent advances of research on deep neural networks provide new tools to build generative models. For example, the attention mechanism can help determine what content in a sentence should be asked [28], and the sequence-to-sequence [2, 7] and the long short-term memory [44] mechanisms are used to generate words to form a question (see, e.g., [10–12, 43]). 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 [48, 53, 54]. However, as pointed out by Kim et al. [16], 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 [9, 20, 37, 49, 51, 55]. For example, Kettip et al. [17] 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. [26] finetuned the pretrained GPT-2 [38] transformer without using any additional complex components or features to enhance its performance. Chen [6] described a fully transformer-based reinforcement learning generator evaluator architecture to generate questions.

In particular, the recent introduction of T5 [39] 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 [38]) 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 a pretrained T5 model, Lidiya et al. [30] combined nine question-answering datasets to finetune a single T5 model and evaluated generated questions using a new semantic measure called BERTScore [52]. Their method achieves so far the best results. We present a finetuned T5 model on a single SQuAD dataset to produce better results.

3 FINETUNING T5

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 finetuning T5, including RACE [19], CoQA [41], and SQuAD [40]. 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, most of which are 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 following format as input:

$$< answer > answer\_text < context > context\_text,$$

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 [27] 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 [47], 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 constant learning rate from 10^{-4} to 10^{-5} with a 2.5×10^{-5} decrement for each model. Likewise, we finetune T5-Large with a constant learning rate from 10^{-5} to 10^{-6} with a 2×10^{-6} decrement for each model.

Evaluated using BLEU [34], ROUGE [22], METEOR [3] and BERTScore [52], 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. The boldfaced number indicates the best in its column. It can be seen 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. Moreover, T5-Large-SQuAD_1 ith the learning rate of 6×10^{-6} offers the second best performance.

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 will also denote T5-Base-SQuAD_1 by T5-SQUAD_1 as in Section 5 when there is no confusion of what size of the dataset is used to pretrain T5.

4 DESCRIPTION OF T5P3

Fig. 1 depicts the architecture of T5P3. The processing pipelines consist of preprocessing to select appropriate answers, question generation, and postprocessing to filter undesirable questions. These

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

Learning Rate	BLEU	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-Lsum	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

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

Learning Rate	BLEU	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-Lsum	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

pipelines also apply to other TP3 models when the underlying transformer is changed to a different one.

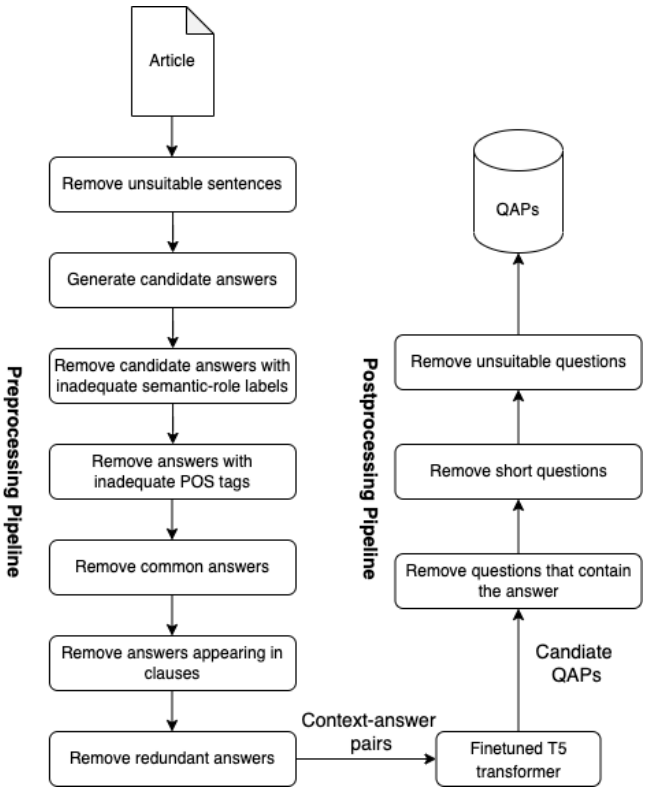

Figure 1: T5P3 Architecture

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 question word in {what, which, when, where, who, whom, whose, why, whether, how} or a question verb, as well as any sentence that ends with a question mark. We then use semantic-role labeling [46] 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 named entity [35] 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 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 function word or the word is used to describe or modify verbs, especially, the word with a POS tag in {RB, RP, CC, DT, IN, MD, PDT, PRP, WP, WDT, WRB} [45] 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 [29]. 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) *Filter 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 [50]. 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 sentence, 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) *Remove 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:

$$< answer > answer\_text < context > context\_text$$

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 [25, 42], 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 postprocessing 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:

Table 3: Automatic evaluation results

Model	Size	BLEU	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-Lsum	METEOR	BERTScore	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**

Parse the answer to constituency tree [14] 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 T5P3-Base (i.e., TP3 based on finetuned T5-Base) and T5P3-Large, we would need to use human judgments. On the other hand, we may compare T5-SQuAD_1 (i.e., without the preprocessing and postprocessing pipelines) with the existing models on QAP generation including the state-of-the-art results under the standard performance metrics over the SQuAD dataset.

5.1 Automatic evaluations

We 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 [36], BART [21], T5 [39] and MixQG [31]. 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 [5].

The results of MixQG_1 were presented in the original paper [31], and the results of MixQG_2 were computed by us using the pretrained model posted on HuggingFace at 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 T5P3

A number of publications (e.g., see [4, 24, 32]) 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 T5P3. We do so on the Gaokao-EN as dataset consisting of 75 articles, where each article contains 15 to 20 sentences. We chose Gaokao-EN because expert evaluations are available from a project we work on. Table 4 depicts the evaluation results. Title abbreviations are

Table 4: Manual evaluation results for T5P3-Base and T5P3-Large over Gaokao-EN

T5P3	Learning Rate	Total QAPs	ADQT QAPs	IA-MLI QAPs	Unusable QAPs	ADQT Ratio (%)	ACPT Ratio (%)
Base	3e-5	**1296**	1036	116	144	79.94	88.89
Large	3e-5	1288	1051	113	124	81.60	90.37
	1e-5	1268	1059	94	115	83.52	90.93
	8e-6	1269	1049	109	111	82.66	91.25
	6e-6	1271	**1063**	112	**96**	**83.63**	**92.45**
	Dynamic	1283	924	136	223	72.02	82.62

explained below, where the numbers in boldface are the best in the corresponding columns:

(1) **Total QAPs** means the total number of QAPs generated by T5P3.
(2) **ADQT QAPs** means the total number of adequate QAPs. Such QAPs can be directly used in applications without modifications.
(3) **IA-MLI QAPs** means inadequate (IN) QAPs where the question, while being semantically correct, contains only a minor language issue (MLI) 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) **Unusable QAPs** means QAPs that don't make any sense.
(5) **ADQT Ratio** means the ratio of the ADQT QAPs over all generated QAPs.
(6) **ACPT Ratio** means the ratio of the ADQT and IA-MLI QAPs over all generted QAPs.

It can be seen that all models generate about the same number of QAPs. Among the models in comparisons, T5P3-Large-6e-6 offers the best performance on both ADQT Ratio and ACPT Ratio. Moreover, T5P3-Large-1e-5 provides the second best performance on the ADQT Ratio, while T5P3-Large-8e-6 offers the second best performance on the ACPT Ratio. T5P3-Large-3e-5 is superior to T5P3-Base-3e-5 on both ratios but T5P3-Base-3e-5 generates the largest number of QAPs.

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/T5P3. The Gaokao-EN dataset and the human judgments of QAPs are 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. Finally, we observe that, while T5P3-Large-6e-6 is in general better than the other models, it's not always the case. For some answers, the questions generated by some other models are better. It would be interesting to figure out how to select the best question generated by different models. For example, we may consider contextual similarity and check correctness of grammars. These issues deserve further investigations.

REFERENCES

[1] Husam Ali, Yllias Chali, and Sadid A Hasan. 2010. Automation of question generation from sentences. In *Proceedings of QG2010: The Third Workshop on Question Generation*. 58–67.
[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).
[3] Satanjeev Banerjee and Alon Lavie. 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*. Association for Computational Linguistics, Ann Arbor, Michigan, 65–72. https://www.aclweb.org/anthology/W05-0909
[4] Chris Callison-Burch, Miles Osborne, and Philipp Koehn. 2006. Re-evaluating the Role of Bleu in Machine Translation Research. In *11th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Trento, Italy, 249–256. https://aclanthology.org/E06-1032
[5] Ying-Hong Chan and Yao-Chung Fan. 2019. A recurrent BERT-based model for question generation. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*. 154–162.
[6] Yu Chen, Lingfei Wu, and Mohammed J. Zaki. 2020. Reinforcement Learning Based Graph-to-Sequence Model for Natural Question Generation. *ArXiv* abs/1908.04942 (2020).
[7] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 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)*. Association for Computational Linguistics, Doha, Qatar, 1724–1734. https://doi.org/10.3115/v1/D14-1179
[8] Guy Danon and Mark Last. 2017. A syntactic approach to domain-specific automatic question generation. *arXiv preprint arXiv:1712.09827* (2017).
[9] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified Language Model Pre-training for Natural Language Understanding and Generation. In *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*.
[10] Xinya Du, Junru Shao, and Claire Cardie. 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)*. Association for Computational Linguistics, Vancouver, Canada, 1342–1352. https://doi.org/10.18653/v1/P17-1123

[11] Nan Duan, Duyu Tang, Peng Chen, and Ming Zhou. 2017. Question generation for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark, 866–874. https://doi.org/10.18653/v1/D17-1090

[12] Vrindavan Harrison and Marilyn Walker. 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. https://doi.org/10.18653/v1/w18-6536

[13] Michael Heilman and Noah A Smith. 2009. *Question generation via overgenerating transformations and ranking*. Technical Report. CARNEGIE-MELLON UNIV PITTSBURGH PA LANGUAGE TECHNOLOGIES INST.

[14] V. Joshi, Matthew E. Peters, and Mark Hopkins. 2018. Extending a Parser to Distant Domains Using a Few Dozen Partially Annotated Examples. In *ACL*.

[15] Payal Khullar, Konigari Rachna, Mukul Hase, and Manish Shrivastava. 2018. Automatic question generation using relative pronouns and adverbs. In *Proceedings of ACL 2018, Student Research Workshop*. 153–158.

[16] Yanghoon Kim, Hwanhee Lee, Joongbo Shin, and Kyomin Jung. 2019. Improving neural question generation using answer separation, In Association for the Advancement of Artificial Intelligence (AAAI). *Proceedings of the AAAI Conference on Artificial Intelligence* 33, 6602–6609. https://doi.org/10.1609/aaai.v33i01.33016602

[17] Kettip Kriangchaivech and Artit Wangperawong. 2019. Question Generation by Transformers. https://doi.org/10.48550/ARXIV.1909.05017

[18] Igor Labutov, Sumit Basu, and Lucy Vanderwende. 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)*. Association for Computational Linguistics, Beijing, China, 889–898. https://doi.org/10.3115/v1/P15-1086

[19] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding Comprehension Dataset From Examinations. *arXiv preprint arXiv:1704.04683* (2017).

[20] Adam D. Lelkes, Vinh Q. Tran, and Cong Yu. 2021. Quiz-Style Question Generation for News Stories. In *Proceedings of the Web Conference 2021 (WWW '21)*. Association for Computing Machinery, New York, NY, USA, 2501–2511. https://doi.org/10.1145/3442381.3449892

[21] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 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*. Association for Computational Linguistics, Online, 7871–7880. https://doi.org/10.18653/v1/2020.acl-main.703

[22] Chin-Yew Lin. 2004. ROUGE: a package for automatic evaluation of summaries. In *Text Summarization Branches Out*. Association for Computational Linguistics, Barcelona, Spain, 74–81. https://www.aclweb.org/anthology/W04-1013

[23] David Lennart Lindberg. 2013. *Automatic question generation from text for self-directed learning*. Ph. D. Dissertation. Applied Sciences: School of Computing Science.

[24] Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 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*. Association for Computational Linguistics, Austin, Texas, 2122–2132. https://doi.org/10.18653/v1/D16-1230

[25] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *ArXiv* abs/1907.11692 (2019).

[26] Luis Enrico Lopez, Diane Kathryn Cruz, Jan Christian Blaise Cruz, and Charibeth Ko Cheng. 2020. Transformer-based End-to-End Question Generation. *ArXiv* abs/2005.01107 (2020).

[27] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*. https://openreview.net/forum?id=Bkg6RiCqY7

[28] Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, 1412–1421. https://doi.org/10.18653/v1/D15-1166

[29] Jean-Baptiste Michel, Yuan Kui Shen, Aviva Presser Aiden, Adrian Veres, Matthew K. Gray, null null, Joseph P. Pickett, Dale Hoiberg, Dan Clancy, Peter Norvig, Jon Orwant, Steven Pinker, Martin A. Nowak, and Erez Lieberman Aiden. 2011. Quantitative Analysis of Culture Using Millions of Digitized Books. *Science* 331, 6014 (2011), 176–182. https://doi.org/10.1126/science.1199644 arXiv:https://www.science.org/doi/pdf/10.1126/science.1199644

[30] Lidiya Murakhovs'ka, Chien-Sheng Wu, Tong Niu, Wenhao Liu, and Caiming Xiong. 2021. MixQG: Neural Question Generation with Mixed Answer Types. https://doi.org/10.48550/ARXIV.2110.08175

[31] Lidiya Murakhovs'ka, Chien-Sheng Wu, Tong Niu, Wenhao Liu, and Caiming Xiong. 2021. MixQG: Neural Question Generation with Mixed Answer Types. arXiv:2110.08175 [cs.CL]

[32] Preksha Nema and Mitesh M. Khapra. 2018. Towards a Better Metric for Evaluating Question Generation Systems. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, 3950–3959. https://doi.org/10.18653/v1/D18-1429

[33] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics* (Philadelphia, Pennsylvania) *(ACL '02)*. Association for Computational Linguistics, USA, 311–318. https://doi.org/10.3115/1073083.1073135

[34] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, 311–318. https://doi.org/10.3115/1073083.1073135

[35] Matthew E. Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. Semi-supervised sequence tagging with bidirectional language models. In *ACL*.

[36] Weizhen Qi, Yu Yan, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. 2020. 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*. 2401–2410.

[37] Weizhen Qi, Yu Yan, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. 2020. ProphetNet: Predicting Future N-gram for Sequence-to-SequencePre-training. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, Online, 2401–2410. https://doi.org/10.18653/v1/2020.findings-emnlp.217

[38] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.

[39] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. , 67 pages. http://jmlr.org/papers/v21/20-074.html

[40] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2383–2392. https://doi.org/10.18653/v1/d16-1264

[41] Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A Conversational Question Answering Challenge. arXiv:1808.07042 [cs.CL]

[42] Nils Reimers and Iryna Gurevych. 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. https://arxiv.org/abs/1908.10084

[43] Mrinmaya Sachan and Eric Xing. 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)*. Association for Computational Linguistics, New Orleans, Louisiana, 629–640. https://doi.org/10.18653/v1/N18-1058

[44] Hasim Sak, Andrew W. Senior, and Françoise Beaufays. 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*.

[45] Beatrice Santorini. 1990. *Part-of-speech tagging guidelines for the Penn Treebank Project*. Technical Report MS-CIS-90-47. Department of Computer and Information Science, University of Pennsylvania.

[46] Peng Shi and Jimmy J. Lin. 2019. Simple BERT Models for Relation Extraction and Semantic Role Labeling. *ArXiv* abs/1904.05255 (2019).

[47] Leslie N. Smith. 2017. Cyclical Learning Rates for Training Neural Networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 464–472. https://doi.org/10.1109/WACV.2017.58

[48] Linfeng Song, Zhiguo Wang, Wael Hamza, Yue Zhang, and Daniel Gildea. 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)*. Association for Computational Linguistics, New Orleans, Louisiana, 569–574. https://doi.org/10.18653/v1/N18-2090

[49] Dan Su, Yan Xu, Wenliang Dai, Ziwei Ji, Tiezheng Yu, and Pascale Fung. 2020. Multi-hop Question Generation with Graph Convolutional Network. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, Online, 4636–4647. https://doi.org/10.18653/v1/2020.findings-emnlp.416

[50] Andrea Varga and L An Ha. 2010. WLV: a question generation system for the QGSTEC 2010 Task B. *Boyer & Piwek (2010)* (2010), 80–83.

[51] Shiyue Zhang and Mohit Bansal. 2019. Addressing Semantic Drift in Question Generation for Semi-Supervised Question Answering. In *EMNLP*.

[52] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. In *International Conference on Learning Representations*. https://openreview.net/forum?id=SkeHuCVFDr

[53] Yao Zhao, Xiaochuan Ni, Yuanyuan Ding, and Qifa Ke. 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*. Association for Computational Linguistics, Brussels, Belgium, 3901–3910. https://doi.org/10.18653/v1/D18-1424

[54] Qingyu Zhou, Nan Yang, Furu Wei, Chuanqi Tan, Hangbo Bao, and Ming Zhou. 2018. Neural question generation from text: a preliminary study. In *Natural Language Processing and Chinese Computing*, Xuanjing Huang, Jing Jiang, Dongyan Zhao, Yansong Feng, and Yu Hong (Eds.). Springer International Publishing, Cham, 662–671.

[55] Wenjie Zhou, Minghua Zhang, and Yunfang Wu. 2019. Question-type Driven Question Generation. In *EMNLP*.