Quiz Design Task:
Helping Teachers Create Quizzes with Automated Question Generation

Philippe Laban  Chien-Sheng Wu  Lidiya Murakhovs’ka
Wenhao Liu  Caiming Xiong
Salesforce AI Research
{plaban, wu.jason, l.murakhovska, wenhao.liu, cxiong}@salesforce.com

Abstract

Question generation (QGen) models are often evaluated with standardized NLG metrics that are based on n-gram overlap. In this paper, we measure whether these metric improvements translate to gains in a practical setting, focusing on the use case of helping teachers automate the generation of reading comprehension quizzes. In our study, teachers building a quiz receive question suggestions, which they can either accept or refuse with a reason. Even though we find that recent progress in QGen leads to a significant increase in question acceptance rates, there is still large room for improvement, with the best model having only 68.4% of its questions accepted by the ten teachers who participated in our study. We then leverage the annotations we collected to analyze standard NLG metrics and find that model performance has reached projected upper-bounds, suggesting new automatic metrics are needed to guide QGen research forward.

1 Introduction

Question generation is a text generation task with practical applications in several settings such as asking clarification questions in dialogue systems (Braslavski et al., 2017), recommending questions during a reading session (Laban et al., 2020), or other educational scenarios such as creating quizzes to emphasize core concepts and engage learners through interaction (Kurdi et al., 2020; Steuer et al., 2021).

The most common automatic evaluation of QGen borrows from other NLG tasks, using metrics such as BLEU (Papineni et al., 2002) to compare system-generated questions with held-out human-written references in terms of n-gram overlap (Amidei et al., 2018). Although they are straightforward to compute, these metrics have been shown to correlate weakly with human opinion in NLG (Gatt and Krahmer, 2018), do not provide a ceiling performance, or insights into the types of errors prevalent in generated questions.

Some prior work has proposed automatic metrics that are specific to QGen, however the metrics are either rule-based (Nema and Khapra, 2018), matching for the presence of certain elements in generated question with limited flexibility, or shown not be beneficial when used to optimize a QGen model through Reinforcement Learning, according to human raters (Hosking and Riedel, 2019).

In this paper, we propose to evaluate QGen with the help of teachers through the Quiz Design Task, illustrated in Figure 1. Human teachers are tasked with creating reading comprehension quizzes for hypothetical students, and QGen models interactively suggest quiz questions which can be accepted or rejected by the teachers. Model performance is tied to the acceptance rate of each model, in other words, the best QGen model is the one with the largest proportion of accepted questions.

There are several definitions for QGen, from clarification question generation (Rao and Daumé III,
2 Quiz Design Task

We propose to evaluate QGen models by measuring how helpful they are for quiz creation. Teachers often have experience with carefully crafting quiz questions, and possess knowledge as to what makes a quality question for a quiz (Pearson and Gallagher, 1983; Kendeou et al., 2016). Meanwhile, they are for the most part unfamiliar with recent progress in language modeling, and do not necessarily know of the limitations of deep learning-based text generation. Therefore they can act as impartial judge in this particular setting in verifying whether question generation models have reached a level at which they can be used to facilitate reading comprehension quiz creation.

2.1 Task Definition

Teachers with experience in designing quizzes are invited to use a quiz design interface (Figure A1), and follow the steps illustrated in Figure 1. They begin by selecting a quiz topic, such as the history of the Statue of Liberty in Figure 1. The system loads reading material relevant to the topic, which can be sourced from a textbook or Wikipedia.

The objective for the teacher is to leverage the reading material and automated QGen models to design an entire quiz composed of 8-12 questions. The teachers proceed by selecting a quiz concept, such as an entity, phrase, or keyword they wish to probe students on. Each evaluated QGen model then generates a candidate question given the entire reading material and the selected quiz concept.

After receiving candidate questions from the QGen models, teachers review and pick which to include in the quiz. Importantly, candidate questions are anonymized and presented in a shuffled order. It is possible that several QGen models generate identical candidates, so we deduplicate the candidates before presenting them to annotators.

Existing question answering human evaluation design either automatically select quiz concepts or answers and questions are evaluated by distinct crowd-workers (Du et al., 2017; Trischler et al., 2017). In the case of Quiz Design Task, we believe that it is important to enable teachers to select quiz concepts themselves, as it allows them to have specific learning objectives, permitting them to assess generated questions with this context in mind.

2.2 Question Error Categorization

To understand model performance beyond overall acceptance rates and assess model limitations, annotators were made to select a reason for each rejected question. However, unlike other NLG tasks, QGen does not have an established error categorization. Therefore, we carried out a formative

---

1https://github.com/salesforce/QGen
study to construct a reusable error categorization for QGen. We collected questions by sampling the QGen models used in the study, and gradually constructed the categorization by labeling and refining the annotations on 976 generated questions. The final categorization is illustrated in Figure 2.

The QGen error categorization we propose is hierarchical, with errors falling in three nested categories. First, similar to the MQM categorization (Lommel et al., 2014) used for translation, the question can be rejected because it is disfluent for example with errors in grammar or repetition. Second, if the question is fluent, it can be rejected for being off target: the answer to the generated question is not the target answer originally selected. Third, if the question is fluent and on target, it can be rejected for being wrong in context (wrong context), for example by being too specific to be natural or not specific enough to be self-contained. Examples of question errors in each category in Table A1.

3 Quiz Setup and Results

3.1 Participant Recruitment

We recruit teachers or ex-teachers from an online group forum. In total, 20 participants filled out an interest form, 14 were selected, and 10 completed the study (with the other 4 either forgetting to complete the task, or completing it partially). The participants had been teachers for at least a year and 3.6 years on average, and had taught diverse subjects such as sciences, history, literature, and IT topics, at various levels from primary school to college-level. The study was meant to last a maximum of two hours, and participants were gifted a $50 gift card upon completion.

The study session began with a tutorial on the interface (see Appendix B) and detailed examples of the error categories. Participants could then clarify any detail before commencing annotation.

3.2 Quiz Topic Selection

Participants were tasked with creating between 5-7 quizzes, each with a minimum of 8 concepts, and could pick from a set list of 7 quiz topics, which we pre-selected from the list of featured Wikipedia articles. We purposefully selected articles within different domains to benchmark the QGen models in diverse topical settings: two in physics (Sustainable Energy, Californium Atom), two in biology (DNA, Enzymes), two in history (Statue of Liberty, Palazzo Pitti), and one in geology (the K-T extinction). Participants were given the first 500 words of the Wikipedia page of each topic as reading material to select Quiz concepts from.

3.3 QGen Models Evaluated

We include seven QGen models of varying size and architecture in our study. First, we finetune three GPT2 baselines (Radford et al., 2019) on the SQuAD dataset (Rajpurkar et al., 2016): GPT2-distil (Sanh et al., 2019), GPT2-base and GPT2-medium. We further add two BART-based (Lewis et al., 2020) models trained on SQuAD as well: BART-base and BART-large. Finally, we include two recent QGen top-performers, ProphetNet (Qi et al., 2020) and MixQG-L (Murakhovs’ka et al., 2022). We limit ourselves to seven models, and exclude larger models (such as GPT2-XL and MixQG-3b) to maintain an interface latency of under 200ms and limit burden to users (Miller, 1968). Details on model training and usage in Appendix A.

3.4 Annotated Results

In total, the study participants annotated 3,164 questions, with 52% of them accepted into a quiz. The distribution of errors per model is summarized in Figure 3. As expected, model size has an effect on performance, with the largest model MixQG-L achieving the highest performance with an acceptance rate of 68.4%, which is more than double the 33.4% achieved by Distil-GPT2.

Almost all models have the largest portion of errors coming from the Wrong Context category. In fact, model improvement mostly comes from...
the other two categories of errors, with a decrease of 40-80% in numbers of errors made in the Disfluent and Off Target categories. In contrast, the MixQG model still generates a Wrong Context question 16.2% of the time, a modest decrease from Distil-GPT2’s 22.3%.

As expected, the Wrong Context category is the most challenging: models have learned to generate fluent questions that are answered by a desired target concept, and still struggle with phrasing the question in a fashion adequate to the context.

4 Analysis

With the annotations collected, we calculate inter-annotator agreement and use the data to benchmark commonly-used NLG metrics.

4.1 Inter-Annotator Agreement

Even though we allow teachers to select their own quiz concepts, in 95 cases, two or more annotators selected the same concept and annotated an identical set of seven candidate questions. This leads us to have a total of 665 questions on which we can compute inter-annotator agreement. On this subset, we measure a Pearson correlation coefficient (Benesty et al., 2009) of 0.47 which can be interpreted as moderate inter-annotator agreement (Schober et al., 2018).

When breaking down the analysis by model origin, the two lowest-performing models (Distil-GPT2 and GPT2-base) obtain the highest agreement rates (above 0.6), showing a stronger agreement on low-quality questions. Notably, ProphetNet obtained the lowest agreement level (0.26). Further investigation reveals that it is the only model generating questions in lowercase. Because our guidelines did not specify how to deal with improper capitalization, some annotators labeled lower-cased questions as a fluency error. This further explains why ProphetNet generated the largest number of disfluent questions. Future work should carefully indicate how to deal with casing and other normalization (such as punctuation) errors.

4.2 Analysis of Existing Metrics

Because several questions for each given context are annotated, we have a unique opportunity to study the commonly-used NLG metrics, and assess which correlate with our annotators’ judgements. We evaluate four of the most commonly used metrics in QGen evaluation: BLEU (Papineni et al., 2002), ROUGE (Lin, 2004) (we include ROUGE-1 and ROUGE-L variants), METEOR (Banerjee and Lavie, 2005), and BERTScore (Zhang et al., 2019). Results are detailed in Table 1.

First, we can use accepted questions as references, and compute metric performance by each system on the dataset we’ve collected. For each metric, we can compute an instance-level correlation (i.e., how well does a metric correlate with annotations for each individual question), as well as system-level correlation (i.e. how similar is the ranking of models according to annotators and according to the metric). As echoed in previous work (Novikova et al., 2017; Chaganty et al., 2018), instance-level correlations are low, but the aggregated metric scores provide high correlation at the system level, with BLEU achieving the highest system-level correlation.

Second, in cases where several questions were marked as acceptable, we can consider each as a valid reference. In such a case, we generate all pairs of references, treating one as a candidate, the other as a reference and computing scores with the standard metrics. The score obtained can be interpreted as an upper-bound for each metric, as they are scores obtained by questions that are judged to all be acceptable.

For all metrics, we find that MixQG has already either surpassed this upper-bound or is within 0.4-1.4 points of doing so. This analysis reveals that even though standard metrics have been useful at measuring progress in NLG, upper-bound performance may be reached soon, and better metrics are needed to guide future progress in QGen and NLG research.

<table>
<thead>
<tr>
<th>Model Name</th>
<th>% Acc.</th>
<th>BLEU</th>
<th>R-I</th>
<th>R-L</th>
<th>MET</th>
<th>BERT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distil-GPT2</td>
<td>33.4</td>
<td>21.2</td>
<td>47.4</td>
<td>45.4</td>
<td>36.8</td>
<td>50.2</td>
</tr>
<tr>
<td>GPT2-base</td>
<td>40.9</td>
<td>26.3</td>
<td>53.1</td>
<td>51.1</td>
<td>43.0</td>
<td>56.1</td>
</tr>
<tr>
<td>GPT2-med</td>
<td>51.3</td>
<td>31.2</td>
<td>57.6</td>
<td>55.4</td>
<td>46.1</td>
<td>59.5</td>
</tr>
<tr>
<td>BART-Base</td>
<td>52.0</td>
<td>31.2</td>
<td>57.2</td>
<td>54.8</td>
<td>46.0</td>
<td>59.9</td>
</tr>
<tr>
<td>ProphetNet</td>
<td>53.5</td>
<td>33.3</td>
<td>62.1</td>
<td>59.3</td>
<td>51.7</td>
<td>57.4</td>
</tr>
<tr>
<td>Bart-Large</td>
<td>58.4</td>
<td>32.4</td>
<td>59.2</td>
<td>56.9</td>
<td>48.8</td>
<td>61.1</td>
</tr>
<tr>
<td>MixQG-L</td>
<td>68.4</td>
<td>33.5</td>
<td>59.6</td>
<td>57.2</td>
<td>50.6</td>
<td>60.0</td>
</tr>
</tbody>
</table>

Table 1: NLG evaluation metrics. For each metric, an upper-bound, and correlations at the instance-level and system-level are computed.

Further investigation reveals that it is the only model generating questions in lowercase. Because our guidelines did not specify how to deal with improper capitalization, some annotators labeled lower-cased questions as a fluency error. This further explains why ProphetNet generated the largest number of disfluent questions. Future work should carefully indicate how to deal with casing and other normalization (such as punctuation) errors.

105
5 Limitations

We now discuss the limitations of the work we’ve presented.

First, even though we attempted to create a realistic scenario in which to evaluate QGen models, some components of the protocol are simplified for practical purposes. For example, the created quiz were not assigned to students, and we rely solely on the teacher’s opinion of the questions as a signal of question quality. Pushing the study further by assigning the quizzes to students and tying question quality to student performance on the quiz would add complexity, but render the protocol more realistic and provide practical learning signals from students.

Second, although we treat teacher annotations as the ground truth, there is some level of disagreement amongst the teachers we recruited, and we measured a moderate level of agreement in Section 4.1. This emphasizes the necessity of thorough and precise guidelines requirements for evaluation protocols, as our lack of rules around the treatment of capitalization of questions led to low agreement on questions generated by an uncased model.

Third, although we gathered a large number of annotations overall, with 3,164 questions annotated in total, this remains small due to the fact that there are many variables on which to break down performance on (e.g., source document, model of origin, annotator). We plan to release the annotation interface as well as the content and models we used to allow future work to expand and reproduce the results.

6 Conclusion

We introduce the Quiz Design task, a human evaluation protocol used to evaluate Question Generation models in an applied scenario. In the QD task, teachers creating a quiz for their students are recommended generated questions, which they can accept in their quiz or reject with a reason from a newly proposed error categorization. We run a QD task with 10 teachers, annotating 3,164 questions originating from seven models, and find that acceptance rates vary widely with the latest QGen models obtaining the highest acceptance rate of 68.4%. Finally, analysis of automatic metrics on our task’s data reveals that even though metrics correlate well with system-level ranks, models have reached potential metric upper-bounds, and improved metrics are required to guide NLG forward.

7 Ethical Considerations

Our experiments were all run for the English language, and even though we expect the study design to be adaptable to other languages, we have not verified this assumption experimentally and limit our claims to the English language. Expanding the claims to other languages would require trained question generation models in the studied language.

The teacher annotators that participated in our study were compensated at a rate above minimum wage, and we have insured that no personally identifiable information is available in the annotations we’ve released.

References


Appendix

A Training Details

We trained five of the QGen models used in the Quiz Design study. They were all trained for ten epochs on the training portion of the SQuAD dataset (Rajpurkar et al., 2016), using the ADAM optimizer (Kingma and Ba, 2015), with hyper-parameter tuning based on model loss on the validation set. The model checkpoint that achieves the lowest validation loss is selected as the final model. Selected hyper-parameters were:

- **Distil-GPT2**: batch-size 32, learning rate $2 \times 10^{-5}$.
- **GPT2-base**: batch-size 32, learning rate $2 \times 10^{-5}$.
- **GPT2-medium**: batch-size 16, learning rate $2 \times 10^{-5}$.
- **BART-base**: batch-size 32, learning rate $1 \times 10^{-4}$.
- **BART-large**: batch-size 32, learning rate $2 \times 10^{-5}$.

Finally, the last two QGen we used are publicly available on the HuggingFace model hub (Wolf et al., 2020), and we use them as is:

- **ProphetNet**: microsoft/prophetnet-large-uncased-squad-qg
- **MixQG**: Salesforce/mixqg-large

With all models, we used beam search to generate candidate questions, using a beam-size of 2, and a sequence length maximum of 30.

B Guidelines to Annotators

We provide the exact guidelines that were given to study participants before they started the annotation procedure:

1. Your objective is to design a quiz about a particular topic for a class of students. The procedure is the following:

2. Select a quiz topic from the list (for example "Sustainable Energy")

3. The system will load a text about the topic.

4. Select a concept that you want to quiz your students on (for example a phrase, a figure, or a keyword) and confirm your selection.

5. **Important**: It is recommended to select shorter concepts, and not full sentences to obtain more precise question. Selecting concepts of up to about 8 words is ideal.

6. The system will load a list of questions that attempt to quiz students about the selected concept.

7. Go over each question, and remove ones you would not include in your quiz. We will next go over types of questions that should be removed.

8. **Important**: you can keep one, multiple or none of the questions (if none of the questions are satisfactory). For each question you remove, you have to choose the reason that the question is unsatisfactory (more on this later).

9. Once you’ve finalized the question for a concept, select another concept and repeat the question selection process. Try to select 8-12 concepts per topic to generate long enough quizzes.

10. Once you’ve finished a full quiz set, you can move on to another quiz topic. We have found that in one hour, you should be able to complete the quizzes for 5 topics.

Following these guidelines, the annotators were provided definitions for each error category, as well as examples similar to the ones shown in Table A1.

C Error Categorization Question Examples

The examples listed in Table A1 were collected during a formative study to establish an error categorization for the task of Question Generation.

D Interface Screenshot

Figure A1 displays a screenshot of the interface used for the Quiz Design Task.
<table>
<thead>
<tr>
<th>Category</th>
<th>Finer Category</th>
<th>Example Question</th>
<th>Rationale</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disfluent</td>
<td>Wrong Tense</td>
<td>What were historically used to disenfranchise racial minorities?</td>
<td>Should be &quot;What was historically...&quot;</td>
</tr>
<tr>
<td></td>
<td>Awkward Phrasing</td>
<td>When did the woolly mammoth die?</td>
<td>Should be &quot;go instinct&quot; rather than &quot;die&quot;</td>
</tr>
<tr>
<td></td>
<td>Not a Question</td>
<td>In January 2020, scientists reported that climate-modeling of the extinction event favors the asteroid impact and not volcanism?</td>
<td>Sentence in declarative format</td>
</tr>
<tr>
<td></td>
<td>Repetition</td>
<td>Who led the team that led the K-Pg boundary clay?</td>
<td>&quot;led&quot; is repeated twice</td>
</tr>
<tr>
<td>Off Target</td>
<td>Unanswerable</td>
<td>Why are DNA studies so important?</td>
<td>Not answered in the DNA Wikipedia page. True answer is Bartholdi, even though target answer was Eiffel (the metalwork builder)</td>
</tr>
<tr>
<td></td>
<td>Other Answer Span</td>
<td>Who designed the Statue of Liberty?</td>
<td></td>
</tr>
<tr>
<td>Wrong Ctxt</td>
<td>Too Specific</td>
<td>Where was the 181 km (114 mi) crater discovered?</td>
<td>Not standard to have unit translations in questions. Question’s target answer is Federal Reserve System</td>
</tr>
<tr>
<td></td>
<td>Reveals Answer</td>
<td>What was the name of the Federal Reserve System? (leading to the creation of the Federal Reserve System)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Inconsistent</td>
<td>What are the only two animals that survived the Cretaceous-Paleogene extinction?</td>
<td>The Wikipedia article mentions species and not animals</td>
</tr>
<tr>
<td></td>
<td>Not Specific Enough</td>
<td>What are some ectothermic species?</td>
<td>Too many ectothermic species are mentioned in the article.</td>
</tr>
</tbody>
</table>

Table A1: **Example generated questions collected during formative study.** These examples form the basis for the error categorization we propose for the QGen task.
Californium is a radioactive chemical element with the symbol Cf and atomic number 98. The element was first synthesized in 1950 at the Lawrence Berkeley National Laboratory (then the University of California Radiation Laboratory), by bombarding curium with alpha particles (helium-4 ions). It is an actinide element, the sixth transuranium element to be synthesized, and has the second-highest atomic mass of all the elements that have been produced in amounts large enough to see with the unaided eye (after einsteinium). The element was named after the university and the U.S. state of California.

Two crystalline forms exist for californium under normal pressure: one above and one below 900 °C (1,650 °F). A third form exists at high...