

Article

QAScore—An Unsupervised Unreferenced Metric for the Question Generation Evaluation

Tianbo Ji ¹ , Chenyang Lyu ^{2,*}, Gareth Jones ¹, Liting Zhou ¹ and Yvette Graham ^{3,*}¹ ADAPT Centre, School of Computing, Dublin City University, 9 Dublin, Ireland² SFI Centre for Research Training in Machine Learning, School of Computing, Dublin City University, 9 Dublin, Ireland³ ADAPT Centre, School of Computer Science and Statistics, Trinity College Dublin, 2 Dublin, Ireland

* Correspondence: chenyang.lyu2@mail.dcu.ie (C.L.); yvette.graham@adaptcentre.ie (Y.G.)

Abstract: Question Generation (QG) aims to automate the task of composing questions for a passage with a set of chosen answers found within the passage. In recent years, the introduction of neural generation models has resulted in substantial improvements of automatically generated questions in terms of quality, especially compared to traditional approaches that employ manually crafted heuristics. However, current QG evaluation metrics solely rely on the comparison between the generated questions and references, ignoring the passages or answers. Meanwhile, these metrics are generally criticized because of their low agreement with human judgement. We therefore propose a new reference-free evaluation metric called QAScore, which is capable of providing a better mechanism for evaluating QG systems. QAScore evaluates a question by computing the cross entropy according to the probability that the language model can correctly generate the masked words in the answer to that question. Compared to existing metrics such as BLEU and BERTScore, QAScore can obtain a stronger correlation with human judgement according to our human evaluation experiment, meaning that applying QAScore in the QG task benefits to a higher level of evaluation accuracy.

Keywords: question generation; question generation evaluation; reference-free evaluation



Citation: Ji, T.; Lyu, C.; Jones, G.; Zhou, L.; Graham, Y. QAScore—An Unsupervised Unreferenced Metric for the Question Generation Evaluation. *Entropy* **2022**, *24*, 1514. <https://doi.org/10.3390/e24111514>

Academic Editor: Adam Lipowski

Received: 12 September 2022

Accepted: 21 October 2022

Published: 24 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Question Generation (QG) commonly comprises automatic composition of an appropriate question given a passage of text and answer located within that text. QG is highly related to the task of machine reading comprehension (MRC), which is a sub-task of question answering (QA) [1–7]. Both QG and MRC receive similar input, a (set of) document(s), while the two tasks diverge on the output they produce, as QG systems generate questions for a predetermined answer within the text while conversely MRC systems aim to answer a prescribed set of questions. Recent QG research suggests that the direct employment of MRC datasets for QG tasks is advantageous [8–10].

In terms of QG evaluation, widely-applied metrics can be categorized into two main classes: word overlap metrics (e.g., BLEU [11] and Answerability [12]) and metrics that employ large pre-trained language models (BLEURT [13] and BERTScore [14]). Evaluation via automatic metrics still face a number of challenges, however. Firstly, most existing metrics are not specifically designed to evaluate QG systems as they are borrowed from other NLP tasks. Since such metrics have been criticized for poor correlation with human assessment in the evaluation of their own NLP tasks such as machine translation (MT) and dialogue systems [15–18], thus that raises questions about the validity of results based on such metrics designed for other tasks. Another challenge lies in the fact that existing automatic evaluation metrics rely on comparison of a candidate with a ground-truth reference. Such approaches ignore the *one-to-many* nature of QG ignoring the fact that a QG system is capable of generating legitimately plausible questions that will be harshly penalised simply for diverging from ground-truth questions. For example, with a passage

describing Ireland, the country located in western Europe, two questions Q1 and Q2, where Q1 = “What is the capital of Ireland?” and Q2 = “Which city in the Leinster province has the largest population?”, can share the same answer “Dublin”. In other words, it is fairly appropriate for a QG system to generate either Q1 or Q2 given the same passage and answer, despite few overlap between the meanings of Q1 and Q2. We deem it the *one-to-many* nature of the QG task, as *one* passage and answer can lead to *many* meaningful questions. A word overlap based metric will however incorrectly assess Q2 with a lower score if it takes Q1 as the reference because of the lack of word overlap between these two questions. A potential solution is to pair each answer with a larger number of hand-crafted reference questions. However, the addition of reliable references requires additional resources, usually incurring a high cost, while attempting to include every possible correct question for a given answer is prohibitively expensive and impractical. Another drawback is that pretrained-model-based metrics require extra resources during the fine-tuning process, resulting in a high cost. Besides the evaluation metrics aforementioned human evaluation is also widely employed in QG tasks. However, the QG community currently lacks a standard human evaluation approach as current QG research employs disparate settings of human evaluation (e.g., expert-based or crowd-sourced, binary or 5-point rating scales) [2,19].

1.1. Contributions

To address the existing shortcomings in QG evaluation, we propose a new automatic metric called QAScore. To investigate whether QAScore can outperform existing automatic evaluation metrics, we additionally devise a new human evaluation approach of QG systems and evaluate its reliability in terms of consistent results for QG systems through self-replication experiments. Details of our contributions are listed as follows:

1. We propose a pretrained language model based evaluation metric called QAScore, which is unsupervised and reference-free. QAScore utilizes the RoBERTa model [20], and evaluates a system-generated question using the cross entropy in terms of the probability that RoBERTa can correctly predict the masked words in the answer to that question.
2. We propose a novel and highly reliable crowd-sourced human evaluation method that can be used as a standard framework for evaluating QG systems. Compared to other human evaluation methods, it is cost-effective and easy to deploy. We further conduct a self-replication experiment showing a correlation of $r = 0.955$ in two distinct evaluations of the same set of systems. According to the results of the human evaluation experiment, QAScore can outperform all other metrics without supervision steps or fine-tuning, achieving a strong Pearson correlation with human assessment;

1.2. Paper Structure

To facilitate the readability, we then introduce the structure of the rest of this paper. Section 2 presents related work about QA and QG tasks, as well as existing evaluation methods, including automatic metrics and human evaluation methods. We additionally provide the comparison between our proposed methods and existing methods. Section 3 describes the methodology and experiment result of the proposed QAScore in detail. Section 4 introduces the design of our newly proposed human evaluation method with the corresponding experiment setting. Finally, Section 5 summarizes the work of this paper, and depicts the future work.

2. Background: Question Answering, Question Generation and Evaluation

2.1. Question Answering

Question Answering (QA) aims to provide answers a to the corresponding questions q . Based on the availability of context c , QA can be categorized into Open-domain QA (without context) [21,22] and Machine Reading comprehension (with context) [23,24]. Besides, QA can also be categorized into generative QA [25,26] and extractive QA [27–30]. Generally, the optimization objective of QA models is to maximize the log likelihood of the ground-truth

answer a for the given context c . Therefore the objective function regarding the parameters θ of QA models is:

$$J(\theta) = \log P(a|c, q; \theta) \quad (1)$$

2.2. Question Generation

Question Generation (QG) is a task where models receive context passages c and answers a , then generate the corresponding questions q which are expected to be semantically relevant to the context c and answers a [5,31]. Thus QG is a reverse/dual task of QA as QA aims to provide answers a to questions q whereas QG targets at generating questions q for the given answers a . Typically, the architecture of QG systems is mainly Seq2Seq model [32] which generates the q word by word in auto-regressive manner. The objective for optimizing the parameters θ of QG systems is to maximize the likelihood of $P(q|c, a)$:

$$J(\theta) = \log P(q|c, a) = \sum_i \log P(q_i|q_{<i}, c, a) \quad (2)$$

2.3. Automatic Evaluation Metrics

We introduce two main categories of automatic evaluation metrics applied for QG task: word-overlap-based metrics and pretrained-model-based metrics in the following sections.

2.3.1. Word-Overlap-Based Metrics

Word-overlap-based metrics usually assess the quality of a QG system according to the overlap rate between the words of a system-generated candidate and a reference. Most of such metrics, including BLEU, GLEU, ROUGE and METEOR are initially proposed for other NLP tasks (e.g., BLEU is for MT and ROUGE is for text summarization), while Answerability is a QG-exclusive evaluation metric.

BLEU Bilingual Evaluation Understudy (BLEU) is a method that is originally proposed for evaluating the quality of MT systems [11]. For QG evaluation, BLEU computes the level of correspondence between a system-generated question and the reference question by calculating the precision according to the number of n -gram matching segments. These matching segments are thought to be unrelated to their positions in the entire context. The more matching segments there are, the better the quality of the candidate is.

GLEU GLEU (Google-BLEU) is proposed to overcome the drawbacks of evaluating a single sentence [33]. As a variation of BLEU, the GLEU score is reported to be highly correlated with the BLEU score on a corpus level. GLEU uses the scores of precision and recall instead of the modified precision in BLEU.

ROUGE Recall-Oriented Understudy for Gisting Evaluation (ROUGE) is an evaluation metric developed for the assessment of the text summarization task, but originally adapted as a recall-adaptation of BLEU [34]. ROUGE-L is the most popular variant of ROUGE, where L denotes the longest common subsequence (LCS). The definition of LCS is a sequence of words that appear in the same order in both sentences. In contrast with sub-strings (e.g., n -gram), the positions of words in a sub-sequence are not required to be consecutive in the original sentence. ROUGE-L is then computed by the F- β score according to the number of words in the LCS between a question and a reference.

METEOR Metric for Evaluation of Translation with Explicit ORdering (METEOR) was firstly proposed to make up for the disadvantages of BLEU, such as lack of recall and the inaccuracy of assessing a single sentence [35]. METEOR first generates a set of mappings between the question q and the reference r according to a set of stages, including: exact token matching (i.e., two tokens are the same), WordNet synonyms (e.g., *well* and *good*), and Porter stemmer (e.g., *friend* and *friends*). METEOR score is then computed by the weighted harmonic mean of precision and recall in terms of the number of unigrams in mappings between a question and a reference.

Answerability Aside from the aforementioned evaluation methods - which are borrowed from other NLP tasks, an automatic metric called Answerability is specifically

proposed for the QG task [12]. Nema and Khapra [12] suggest combining it with other existing metrics since its aim is to measure how answerable a question is, something not usually targeted by other automatic metrics. For example, given a reference question r : “What is the address of DCU?” and two generated questions q_1 : “address of DCU” and q_2 : “What is the address of”, it is obvious that q_1 is rather answerable since it contains enough information while q_2 is very confusing. However, any similarity-based metric is certainly prone to think that q_2 (ROUGE-L:90.9; METEOR:41.4; BLEU-1:81.9) is closer to r than q_1 (ROUGE-L:66.7; METEOR: 38.0; BLEU-1:36.8). Thus, Answerability is proposed to solve such an issue. In detail, for a system-generated question q and a reference question r , the Answerability score can be computed as shown in Equation (3):

$$\begin{aligned} P &= \sum_{i \in E} w_i \frac{h_i(q, r)}{k_i(q)} \\ R &= \sum_{i \in E} w_i \frac{h_i(q, r)}{k_i(r)} \\ \text{Answerability} &= \frac{2 \times P \times R}{P + R} \end{aligned} \quad (3)$$

where i ($i \in E$) represents certain types of elements in $E = \{R, N, Q, F\}$ (R = Relevant Content Word, N = Named Entity, Q = Question Type, and F = Function Word). w_i is the weight for type i that $\sum_{i \in E} w_i = 1$. Function $h_i(x, y)$ returns the number of i -type words in question x that have matching i -type words in question y , and $k_i(x)$ returns the number of i -type words occurring in question x . The final Answerability score is the F1 score of Precision P and Recall R .

Along with using Answerability individually, a common practice is to combine it with other metrics as suggested when evaluating QG systems [6,36]:

$$\text{Metric}_{mod} = \beta \cdot \text{Answerability} + (1 - \beta) \cdot \text{Metric}_{ori} \quad (4)$$

where Metric_{mod} is a modified version of an original evaluation metric Metric_{ori} using Answerability, and β is a hyper-parameter. In this experiment, we combine it with BLEU to generate the Q-BLEU score using the default value of β .

2.3.2. Pretrained-Model-Based Metrics

BERTScore Zhang et al. [14] proposed an automatic metric called BERTScore for evaluating text generation task because word-overlap-based metrics like BLEU fail to account for compositional diversity. Instead, BERTScore computes a similarity score between tokens in a candidate sentence and its reference based on their contextualized representations produced by BERT [37]. Given a question that has m tokens and a question that has n tokens, the BERT model can first generate the representations of q and r as $q = \langle q_1, q_2, \dots, q_m \rangle$ and $r = \langle r_1, r_2, \dots, r_n \rangle$, where q_i and r_i respectively mean the contextual embeddings of the i -th token in q and r . Then, the BERT score between the question and the reference can be computed by Equation (5):

$$\begin{aligned} P_{\text{BERT}} &= \frac{1}{m} \sum_{p_i \in p} \max_{r_j \in r} p_i^\top r_j \\ R_{\text{BERT}} &= \frac{1}{n} \sum_{r_i \in r} \max_{p_j \in p} p_j^\top r_i \\ \text{BERTScore} &= \frac{2 \cdot P_{\text{BERT}} \cdot R_{\text{BERT}}}{P_{\text{BERT}} + R_{\text{BERT}}} \end{aligned} \quad (5)$$

where the final BERTScore is the F1 measure computed by precision P_{BERT} and recall R_{BERT} .

BLEURT BLEURT is proposed to solve the issue that metrics like BLEU may correlate poorly with human judgments [13]. It is a trained evaluation metric which takes a candidate and its reference as input and gives a score to indicate how the candidate

can cover the meaning of the reference. BLEURT uses a BERT-based regression model trained on the human rating data from the WMT Metrics Shared Task from 2017 to 2019. Since BLEURT was proposed for evaluating models on the sentence level, meanwhile no formal experiments are available for corpus-level evaluation, we directly compute the final BLEURT score of a QG system as the arithmetic mean of all sentence-level BLEURT scores in our QG evaluation experiment as suggested (see the discussion on <https://github.com/google-research/bleurt/issues/10>, accessed on 10 June 2021).

2.4. Human Evaluation

Although the aforementioned prevailing automatic metrics mentioned above are widely employed for QG evaluation, criticism of n -gram overlap-based metrics' ability to accurately and comprehensively evaluate the quality has also been highlighted [38]. As a single answer can potentially have a large number of corresponding plausible questions, simply computing the overlap rate between an output and a reference to reflect the real quality of a QG system does not seem convincing. A possible solution is to obtain more correct questions per answer, as n -gram overlap-based metrics would usually benefit from multiple ground-truth references. However, this may elicit new issues: (1) adding additional references over the entire corpora requires similar effort to creating a new data set incurring expensive and time resource costs; (2) it is not straightforward to formulate how word overlap should contribute to the final score for systems.

Hence, human evaluation is also involved when evaluating newly proposed QG systems. A common approach is to evaluate a set of system-generated questions and ask human raters to score these questions on an n -point Likert scale. Below we introduce and describe recent human evaluations are applied to evaluate QG systems.

Jia et al. [39] proposed EQG-RACE to generate examination-type questions for educational purposes. 100 outputs are sampled and three expert raters are required to score these outputs in three dimensions: fluency—*whether a question is grammatical and fluent*; relevancy—*whether the question is semantically relevant to the passage*; and answerability—*whether the question can be answered by the right answer*. A 3-point scale is used for each aspect, and aspects are reported separately without overall performance.

KD-QG is a framework with a knowledge base for generating various questions as a means of data augmentation [40]. For its human evaluation, three proficient experts are individually assigned to 50 randomly-sampled items and judge whether an assigned item is *reliable* on a binary scale (0–1). Any item with a positive reliability will be further assessed for its level of *plausibility* on a 3-point scale (0–2) that is construed as: 0—*obviously wrong*, 1—*somewhat plausible* and 2—*plausible*. These two aspects are treated separately without reporting any overall ranking.

Answer-Clue-Style-aware Question Generation (ACS-QG) aims to generate questions together with the answers from unlabeled textual content [41]. Instead of evaluating the questions alone, a sample is a tuple of (p, q, a) where p = passage, q = question and a = answer. A total of 500 shuffled samples are assigned to ten volunteers, where each volunteer receives 150 samples to ensure an individual sample is evaluated by three different volunteers. Three facets of a sample are evaluated: *well-formedness* (yes/understandable/no)—*if the question is well-formed*; *relevancy* (yes/no)—*if the question is relevant to the passage*; and *correctness* (yes/partially/no)—*if the answer the question is correct*. The results for each facet are reported as percentages rather than as a summarized score.

Ma et al. [42] proposed a neural QG model consisting of two mechanisms: semantic matching and position inferring. The model is evaluated by human raters for three aspects: *semantic-matching*, *fluency*, and *syntactic-correctness* on a 5-point scale. However, the details about: (1) the number of evaluated samples; (2) the number of involved raters; (3) the type of human raters (crowd-sourced or experts) are unfortunately not provided.

QURIOUS is a pretraining method for QG, and QURIOUS-based models are expected to outperform other non-QURIOUS models [43]. To verify this, a crowd-sourced human evaluation experiment is then conducted. Thirty passages with answers are ran-

domly selected, and human raters compare questions from two distinct models. For each single comparison, 3 individuals are involved. Specifically, a human rater is presented with a passage, an answer, and questions *A* and *B* from two models, and is asked to rate which question is better than the other according to two aspects: *naturalness*—the question is fluent and written in well-formed English, and *correctness*—the question is correct given the passage and the answer. Each comparison has one of the three distinct choices: (*A* = *best*, *B* = *worst*), (*A* = *equal*, *B* = *equal*) and (*A* = *worst*, *B* = *best*), and the final human score of a system for each aspect is computed as the number of times it is rated as *best* subtracting the number of times it is rated as *worst*, followed by dividing by the number of times it is evaluated in total.

Although the fact that human evaluation is somewhat prevalent in the QG evaluation much more than many other NLP areas, there still remains three major issues:

1. There still lacks a standard human evaluation for QG since the aforementioned examples individually use disparate rating options and settings with only a few overlaps. These existing methods for the QG task can generally change from one set of experiments to the next, highlighting the lack of a standard approach, making comparisons challenging;
2. The vast majority of QG human evaluation methods are either expert-based or volunteer-based, with the former are normally expensive and latter likely incurring issues such as shortages of rater availability. Furthermore, the inconvenience of deploying human evaluation at scale can lead to a small sample size, which could possibly hinder the reliability of evaluation results;
3. Much of the time, details of human evaluation experiments are vague with on occasion sample sizes and number of raters omitted from publications. Although expert-based human evaluation can be deemed to have a high level of rater's agreement, such information is seldom reported, resulting in difficulties interpreting the reliability and validity of experiments, in particular when crowd-sourced human evaluation is employed.

2.5. Comparison with Existing Evaluation Methods

Our contributions include a new automatic metric called QAScore and a new human evaluation method, and we will compare them with existing methods.

2.5.1. QAScore and Existing Automatic Metrics

The aforementioned evaluation metrics, including both word-overlap-based and pretrained-model-based metrics, generally evaluate a candidate question only using a reference, which unfortunately fail to consider the impacts of given context, while QAScore can evaluate a question together with its passage and answer. Furthermore, QAScore can evaluate a question needing no reference. Compared with other metrics, QAScore additionally correlates better with human judgement. The results of comparison between QAScore and these metrics will be introduced in detail in Section 3.5.

2.5.2. Our Human Evaluation Method with Existing Methods

Compared to the commonly applied expert-based human evaluation method in the QG task, our human evaluation method is a crowd-sourcing method which is deemed to be cheaper. In addition, our method can be easily deployed on a large scale, rendering more evaluation data. Meanwhile, we provide the details of our experiment in Section 4, including the cost, elapsed time and number of involved human rater, while such information is generally vague or unavailable in aforementioned human evaluation methods.

3. QAScore—An Automatic Metric for Evaluating QG Systems Using Cross-Entropy

Since QG systems are required to generate a question according to a passage and answer, we think that the evaluation of the question should take into account the passage and answer as well, which current metrics fail to achieve. In addition, the QG evaluation

should consider the *one-to-many* nature (see Section 1) that there may be many appropriate questions based on one passage and answer, while current metrics usually have only one or a few references to refer to. Furthermore, metrics, such as BERTScore and BLEURT, require extra resources for fine-tuning, which is expensive and inconvenient for utilization. Hence, we propose a new automatic metric, which has three main advantages compared with existing automatic QG evaluation metrics: (1) it can directly evaluate a candidate question with no need to compute the similarity with any human-generated reference; (2) it is easy to deploy as it takes a pretrained language model as the scorer and requires no extra data for further fine-tuning; (3) it can correlate better with humans according to human judgements than other existing metrics.

3.1. Proposed Metric

In this section, we describe our proposed pretrained-model-based unsupervised QG metric. The fact that there are many possible correct questions for the same answer and passage means that multiple distinct questions can legitimately share the same answer, due to the one-to-many nature of QG task as described in Section 4.1. Hence, we think a reference-free metric is more appropriate since there can be several correct questions for a given pair of an answer and a passage. In addition, the QG and QA tasks are complementary where the common practice of QG is to generate more data to augment a QA dataset and QA systems can benefit from the augmented QA dataset [44]. Therefore, we believe a QA system should be capable of judging the quality of the output of a QG system, and the proposed metric is designed to score a QG output in a QA manner whose detail will be introduced in Section 3.2. Furthermore, our proposed metric has the advantage of being unsupervised. Pretrained language models are demonstrated to contain plenty of useful knowledge since they are trained on large scale corpus [45]. Therefore, we plan to directly employ a pretrained language model to act as a evaluation metric without using other training data or supervision, as introduced in Section 3.2.1.

3.2. Methodology

Since QG and QA are two complementary tasks, we can naturally conjecture that a QG-system-generated question can be evaluated according to the quality of the answer generated by a QA system. Therefore, we think the likelihood of the generated question q to the given answer a and passage p should be proportional to the likelihood of the corresponding answer a to the generated question q and passage p :

$$P(q|p, a) \propto P(a|p, q) \quad (6)$$

We take the passage and the answer a , “commander of the American Expeditionary Force (AEF) on the Western Front” (see the example in Figure 3 which will be introduced in Section 4). We show two distinct question q_1 and q_2 , where q_1 is “What was the most famous post of the man who commanded American and French troops against German positions during the Battle of Saint-Mihiel?” and q_2 is “What was the Battle of Saint-Mihiel?”. It can be found that, a is the correct answer to q_1 rather than q_2 . Therefore, in this case a QA model is *more* likely to generate a when given q_1 , and it is expected *not* to generate a when given q_2 . In another words, the likelihood that a QA model can produce a given q_1 is more than that given q_2 :

$$P(q_1|p, a) > P(q_2|p, a) \quad (7)$$

The detailed scoring mechanism will be introduced in Section 3.2.2.

3.2.1. Pre-Trained Language Model–RoBERTa

We chose to employ the masked language model RoBERTa [20] in a MRC manner to examine the likelihood of an answer, and its value can act as the quality of the target question to be evaluated. RoBERTa (**R**obustly **o**ptimized **B**ERT **a**pproach) is a BERT-based approach for pretraining a masked language model. Compared with the original BERT,

RoBERTa is trained on a larger dataset with a larger batch size and longer elapsed time. It also removes the next sentence prediction (NSP) step and leverages full-sentences (sentences that reach the maximal length). For text encoding, RoBERTa employs a smaller BPE (Byte-Pair Encoding) vocabulary from GPT2 instead of the character-level BPE vocabulary used in the original BERT.

In general, the pre-training objective of RoBERTa aims to predict the original tokens which are replaced by a special token [46]. Given a sequence (w_1, w_2, \dots, w_n) , a token w in the original sentence is randomly replaced by a special token $[MASK]$. And the pre-training objective of RoBERTa can be formulated as Equation (8):

$$J(\theta) = \log P(\hat{W}|\tilde{W}) = \sum_{i \in \hat{I}} \log P(w_i | w_{j_1}, w_{j_2}, \dots, w_{j_n}; j_k \in \{I - \hat{I}\}) \quad (8)$$

where \hat{W} and \tilde{W} represent masked words and unmasked words respectively, I denotes the original indices of all tokens including masked and unmasked tokens, \hat{I} represents the indices of masked tokens, and the indices of unmasked tokens can be denoted as $I - \hat{I}$.

3.2.2. Process of Scoring

Since this proposed metric leverages a means of question answering to assess QG-system-generated questions, we call it QAScore. Given the passage, the correct answer, and the QG-system-generated question, we first encode and concatenate the passage and the answer. Figure 1 provides a visualization of the process of scoring a generated question using its passage and answer using the masked language model RoBERTa. First, the passage and the question are concatenated by the end-of-sequence token $\langle eos \rangle$, which represents the context for the masked language model. Next, the masked answer containing one masked word is concatenated by the context together with the $\langle eos \rangle$ token as the input for the model. The model is then asked to predict the real value of the masked word using the context and the masked answer. The likelihood that RoBERTa can generate the true word can act as the score for that masked word. For the evaluation of a single question, all words in the given answer will be masked in a one-at-a-time manner. The final metric score of the question Q can be computed by the following equation:

$$o_w = M(P, Q, A_{\tilde{w}}) \quad (9)$$

$$p_w = \log \frac{e^{o_w}}{\sum_{w'} e^{o_{w'}}} \quad (10)$$

$$l_w = \sum_{c=1}^C y_{w,c} \cdot p_{w,c} \quad (11)$$

$$QAScore(Q) = \sum_{w \in A} l_w \quad (12)$$

where Equation (9) computes the output of the model RoBERTa (M) when receiving the passage P , the question Q , and $A_{\tilde{w}}$ which is the Answer A with a word w in it masked, as the input. Then, Equation (10) computes the probabilities of o_w using the log-softmax function. Equation (11) is the log likelihood of p_w where C is the number of vocabulary size of RoBERTa. Finally, the QAScore of the question Q is the sum of l_w among all words in its relevant Answer A .

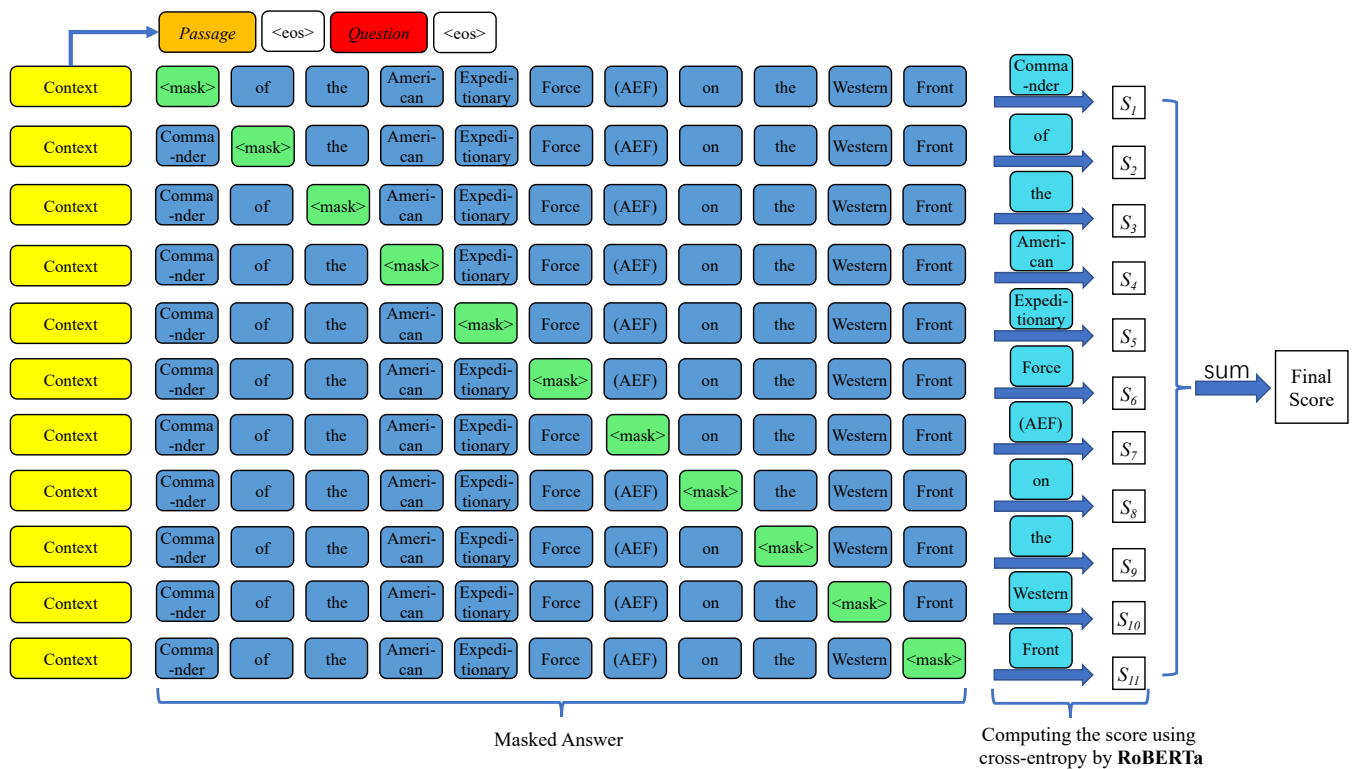


Figure 1. The process of scoring a question by RoBERTa, where the context (yellow) contains the passage and the question (to be evaluated), <eos> is the separator token, the score of a single word is the likelihood that RoBERTa can predict the real word (cyan) which is replaced by the mask token <mask> (green) in the original answer, and the final metric score is the sum of scores of all words in the answer.

3.3. Dataset and QG Systems

HotpotQA Dataset

We conduct the experiment on the HotpotQA dataset [47], initially proposed for the multi-hop question answering task (see <https://hotpotqa.github.io/>, accessed on 10 June 2021). The term, multi-hop, means that a machine should have the ability to answer given questions by extracting useful information from several related passages. The documents in the dataset are extracted from Wikipedia articles, and the questions and answers are created by crowd workers. A worker is asked to provide the questions whose answers requires reasoning over all given documents. Each question in the dataset is associated with one correct answer and multiple passages, where the answer is either a sub-sequence from the passage or simply yes-or-no. These multiple passages are treated as a simple passage to show to human raters during the experiment. Note that the original HotpotQA test set provides no answer for each question, and such a set is inappropriate for the QG task as an answer is necessary for a QG system to generate a question. Instead, a common practice is to randomly sample a fraction from the training set as the validation set, and the original validation set can act as the test set when training or evaluating a QG system based on a QA dataset. The test set we used to grab system-generated outputs for the QG evaluation is in fact the validation set.

Besides, HotpotQA dataset provides two forms of passages: full passages and supporting facts. For each question, its full passages, on the average, consist of 41 sentences while the average number of sentences in its supporting facts is 8. Since the reading quantity is one of our concerns, we use the sentences from supporting facts to constitute the passage to prevent workers from reading too many sentences per assignment.

3.4. QG Systems for Evaluation

To analyze the performance of our proposed evaluation method, 11 systems will be evaluated, including 10 systems that are trained on the HotpotQA dataset and the Human system that can represent the performance of humans on generating questions. The Human system is directly made up of the questions extracted from the HotpotQA testset. The 10 trained systems are from the following neural network models:

- **T5 (small & base)**: a model using a text-to-text transfer transformer that is pretrained on a large text corpus [48];
- **BART (base & large)**: a denoising auto-encoder using the standard sequence-to-sequence transformer architecture [49];
- **Att-GGNN**: an attention-based gated graph neural network model [3];
- **Att-GGNN (plus)**: a variant of Att-GGNN model which is combined with the context switch mechanism [19];
- **H-Seq2seq**: a hierarchical encoding-decoding model proposed for the QG task [19];
- **H-Seq2seq***: a variant of H-Seq2seq which utilizes a larger dictionary for the avoidance of generating the unknown token $\langle \text{UNK} \rangle$;
- **GPT-2**: a large transformer-based language model with parameters reaching the size of 1.5 B [50].
- **RNN**: a sequence-to-sequence model using the vanilla current neural network (RNN) structure [51].

These systems then generate questions on the HotpotQA testset.

3.5. Results

Table 1 shows the human scores (z) and the metric scores of QG systems evaluated using QAScore and current prevailing QG evaluation metrics, where the human score is computed according to our newly proposed human evaluation method, which will be introduced in Section 4. Table 2 describes how these metrics correlate with human judgements according to the results of the human evaluation experiment. Since our metric does not rely on a ground-truth reference, we can additionally include the result of the Human system unlike other automatic metrics. It can be seen that our metric correlates with human judgements at 0.864 according to the Pearson correlation coefficient, where even the best automatic metric METEOR can only reach 0.801 (see Table 2). Also, compared with the other two pretrained-model-based metrics BERTScore and BLEURT, our metric can outperform them at >0.1 . In terms of Spearman, our metric achieves $\rho \approx 0.8$ where other metrics can only reach at most $\rho \approx 0.6$. In addition, our metric also outperforms other metrics according to Kendall's tau since it reach at $\tau \approx 0.7$ and other metrics merely achieve at most $\tau \approx 0.5$. We can conclude that our metric correlates better with human judgements with respect to all three categories of correlation coefficients.

Table 1. QG system evaluation scores, including human scores (z), QAScore, and other existing metrics.

| System | z | QAScore | METEOR | ROUGE-L | BERTScore | BLEURT | Q-BLEU4 | Q-BLEU1 |
|--------------------------|-------|---------|--------|---------|-----------|--------|---------|---------|
| Human | 0.322 | −0.985 | — | — | — | — | — | — |
| BART _{large} | 0.308 | −1.020 | 30.18 | 47.58 | 90.85 | −0.363 | 43.77 | 51.47 |
| BART _{base} | 0.290 | −1.030 | 29.66 | 47.13 | 90.74 | −0.381 | 44.14 | 51.65 |
| T5 _{base} | 0.226 | −1.037 | 27.99 | 41.60 | 88.44 | −0.682 | 37.78 | 44.84 |
| RNN | 0.147 | −1.064 | 15.46 | 26.77 | 84.59 | −1.019 | 9.68 | 15.92 |
| H-Seq2seq | 0.120 | −1.076 | 17.50 | 29.86 | 85.49 | −0.953 | 10.51 | 17.74 |
| T5 _{small} | 0.117 | −1.049 | 23.62 | 32.37 | 86.34 | −0.860 | 26.73 | 32.92 |
| Att-GGNN _{plus} | 0.076 | −1.065 | 21.77 | 36.31 | 86.27 | −0.784 | 12.63 | 19.86 |

Table 1. Cont.

| System | z | QAScore | METEOR | ROUGE-L | BERTScore | BLEURT | Q-BLEU4 | Q-BLEU1 |
|------------|--------|---------|--------|---------|-----------|--------|---------|---------|
| H-Seq2seq* | 0.053 | −1.045 | 18.23 | 31.69 | 85.83 | −0.866 | 11.12 | 18.36 |
| Att-GGNN | −0.008 | −1.068 | 20.02 | 33.60 | 86.00 | −0.802 | 11.13 | 18.67 |
| GPT-2 | −0.052 | −1.108 | 16.40 | 29.98 | 86.44 | −0.899 | 24.83 | 31.85 |

Table 2. The Pearson (r), Spearman (ρ) and Kendall's tau (τ) correlation between automatic metric scores and human judgements.

| | QAScore | METEOR | ROUGE-L | BERTScore | BLEURT | Q-BLEU4 | Q-BLEU1 |
|--------|---------|--------|---------|-----------|--------|---------|---------|
| r | 0.864 | 0.801 | 0.770 | 0.761 | 0.739 | 0.725 | 0.724 |
| ρ | 0.827 | 0.612 | 0.503 | 0.430 | 0.503 | 0.467 | 0.467 |
| τ | 0.709 | 0.511 | 0.378 | 0.289 | 0.378 | 0.289 | 0.289 |

4. New Human Evaluation Methodology

To investigate the performances of QAScore and current QG evaluation metrics, and to overcome the issues described in Section 2, we propose a new human evaluation method for assessing QG systems in this section. First, this can be used as a standard framework for evaluating QG systems because of its flexible evaluation criteria unlike other model-specific evaluation methods. Second, it is a crowd-sourcing human evaluation rather than expert-based, thus it can be deployed on a large scale within an affordable budget. Furthermore, the self-replication experiment proves the robustness of our method, and we provide the specified details of our method and corresponding experiment for reproduction and future studies.

4.1. Experiment Design

In this section, the methodology of our proposed crowd-sourcing human evaluation for QG is introduced. An experiment that investigates the reliability of results for the new method is also provided, as well as details such as the design of interface shown to human raters, mechanisms for quality checking the evaluation, and the evaluation criteria employed. Meanwhile, our experiment will be deployed on the crowd-sourcing platform Amazon Mechanical Turk (AMT) (www.mturk.com, accessed on 10 June 2021), where each task assigned to human workers is called a Human Intelligence Task (HIT).

4.1.1. Methodology

QG receives a context with a sentence as the input and generates a textual sequence as the output, with automatic metrics reporting the computation of word/ n -gram overlap between the generated sequence and the reference question. However, human evaluation can vary. When evaluating MRC systems via crowd-sourced human evaluation, raters are asked to judge system-generated answers with reference to gold standard answers because a correct answer to the given question should be, to some degree, similar to the reference answer [52].

Whereas, simply applying the same evaluation is not ideal since evaluating a QG system is more challenging due to its *one-to-many* nature (see Section 1), namely a QG system can produce a question that is appropriate but distinct from the reference. Such evaluation may unfairly underrate the generated question because of its inconformity with the reference. To avoid this situation in our experiment, we ask a human rater to directly judge the quality of a system-generated question with its passage and answer present, instead of a reference question.

4.1.2. Experiment User Interface

Since our crowd-sourced evaluation method can involve workers who have no specific knowledge of the related field, a minimal level of guidance is necessary to concisely introduce the evaluation task. Prior to each HIT, a list of instructions followed by button labelled *I understand* is provided, with the human rater beginning a HIT by clicking the button. The full list of instructions is described in Figure 2. In regard to the fourth instruction, Chrome browser is recommended to ensure the stability because we present the HTML element “range control” embedded with hashtags while not all browsers can fully support this feature (e.g., Firefox does not support this feature at all).

-
1. Each time you will see a question together with a passage, and your task is to rate the questions after reading the given passage.
 2. Each HIT contains one certain passage with 20 various questions to rate.
 3. The highlighted content in the passage is expected to be the answer to the presented question. A passage with no highlighted content means the question should be a “yes-or-no” question.
 4. Chrome is preferred, other browsers may cause some errors.
 5. There is a feedback box at the end of the HIT. If you encounter any problems, please enter them in this box or email our MTurk account.
-

Figure 2. Full instructions shown to a crowd-sourced human assessor read prior to starting HITs.

Within each HIT, a human assessor is required to read a passage and a system-generated question with the input (correct) answer, then rate the quality of the question according to the given passage and the answer. Since the answer is a sub-sequence of the passage, we directly emphasize the answer within the passage. Figure 3 provides an example of the interface employed in experiments, where a worker is shown a passage whose highlighted contents are expected to be the answer to the generated question. Meanwhile, workers may see a passage without any highlighted content since a fraction of the answers are simply “yes-or-no”.

| | |
|------------------|---|
| Passage: | The Battle of Saint-Mihiel was a major World War I battle fought from 12–15 September 1918, involving the American Expeditionary Force (AEF) and 110,000 French troops under the command of General John J. Pershing of the United States against German positions. The U.S. Army Air Service (which later became the U.S. Air Force) played a significant role in this action. General of the Armies John Joseph “Black Jack” Pershing (13 September 1860–15 July 1948) was a senior United States Army officer. His most famous post was when he served as the commander of the American Expeditionary Force (AEF) on the Western Front in World War I, 1917–1918. |
| Question: | What was the most famous post of the man who commanded American and French troops against German positions during the Battle of Saint-Mihiel? |

Figure 3. The interface shown to human workers, including a passage with highlighted contents and a system-generated question. The worker is then asked to rate the question.

4.1.3. Evaluation Criteria

Human raters assess the system output question in regards to a range of different aspects (as opposed to providing a single overall score). Figure 4 provides an example rating criterion, where a human rater is shown a Likert statement and asked to indicate the level of agreement with it through a range slider from *strongly disagree* (left) to *strongly agree* (right).

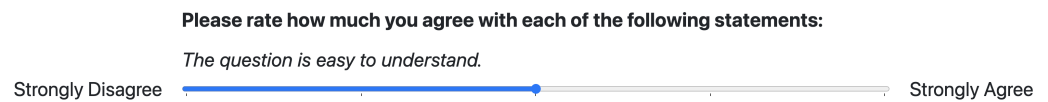


Figure 4. The example of a Likert statement of an evaluation criterion shown to a human worker.

The full list of evaluation criteria we employed in this experiment is available in Table 3, where the labels are in reality not shown to the workers during the evaluation. As an empirical evaluation method, these criteria are those most commonly employed in current research but can be substituted for distinct criteria if needed (see Section 2.4). Since our contribution focuses on proposing a human evaluation approach that can act as a standard for judging QG systems, rather than proposing a fixed combination of evaluation criteria, the criteria we employed are neither immutable nor hard-coded. And we encourage adjusting, extending and pruning them if necessary. Additionally, the rating criterion “answerability” in Table 3 should not be confused with the automatic metric Answerability.

Table 3. The rating criteria of assessing the quality of a system-generated question. Note that only the Likert statements are available for human workers and the labels are not shown in the experiment.

| Label | Likert Statement |
|--------------------------|---|
| <i>Understandability</i> | <i>The question is easy to understand.</i> |
| <i>Relevancy</i> | <i>The question is highly relevant to the content of the passage.</i> |
| <i>Answerability</i> | <i>The question can be fully answered by the passage</i> |
| <i>Appropriateness</i> | <i>The question word (where, when, how, etc.) is fully appropriate.</i> |

4.2. Quality Control

Similar to human evaluation experiments in other tasks (e.g., MT and MRC) [53], quality-controlling the crowd-sourced workers is likewise necessary for the QG evaluation. Since no ground-truth reference will be provided for the comparison with system-generated questions, the quality control methods involve no “reference question”. Two methods—*bad reference* and *repeat*—are employed the means of quality-controlling the crowd to filter out incompetent results.

Bad reference: A set of system-generated questions are randomly selected, and their degraded versions are automatically generated to make a set of bad references. To create a *bad reference* question, we took the original system-generated question and degraded its quality by replacing a random short sub-string from it with another string. The replacement samples are extracted from the entire set passages and should have the same length with the replaced string. Given the original question that consists of n words, the number of words that the replacement should have is subsequently decided on the following rules:

- for $1 \leq n \leq 3$, it comprises 1 word.
- for $4 \leq n \leq 4$, it comprises 2 words.
- for $6 \leq n \leq 8$, it comprises 3 words.
- for $9 \leq n \leq 15$, it comprises 4 words.
- for $16 \leq n \leq 20$, it comprises 5 words.
- for $n \geq 21$, it comprises $\lfloor n/5 \rfloor$ words.

Initial and final words are not included for questions with more than two words, and the passage regarding the current question is also excluded.

Repeat: a set of system-generated questions are randomly selected, and they are copied to make a set of repeats.

In order to implement quality control, we will apply a significance test between the paired bad references and their associate ordinary questions on all rating types. In this case, a non-parametric paired significance test, Wilcoxon signed-rank test, is utilized as we cannot assume the scores are normally distributed. We use two set $Q = \{q_1, q_2, \dots\}$ and $B = \{b_1, b_2, \dots\}$ to represent the ratings of ordinary questions and bad references, where q_i

and b_i respectively represent the scores of n rating criteria for an ordinary question and its related bad reference. For this experiment, we have 4 rating criteria as described in Table 3. We then compare the p -value produced by the significant test between Q and B with a selected threshold α to test whether the scores of ordinary questions are significantly higher than those of bad references. We apply the significance test on each worker, and the HITs from a worker with resulting $p < \alpha$ are kept. We choose $\alpha = 0.05$ as our threshold as it is a common practice [52,53].

Structure of HIT

Figure 5 demonstrates the HIT structure in our human evaluation experiment, where ORD = ordinary question, REPEAT = repeat question and BADREF = bad reference question. Each HIT consists of 20 rating items, including: (a) 11 ordinary system-generated questions; (b) 6 bad reference questions corresponding to 6 of these 11; (c) 3 exact repeats corresponding to 3 of these 11. We organize a single HIT as follows:

- 1 original question, 1 repeat and 1 bad reference from the Human system (comprising a total of 3 questions);
- 2 original questions and their repeats from 2 of the 10 neural QG systems (comprising a total of 4 questions);
- 5 original questions and their bad references from the other 5 of the 10 normal systems (comprising a total of 10 questions);
- 3 original questions from the remaining 3 of the 10 normal systems (comprising a total of 3 questions).

Although the hierarchical structure in Figure 5 seems to organize the 20 items in a certain order, they will be fully shuffled before the deployment.

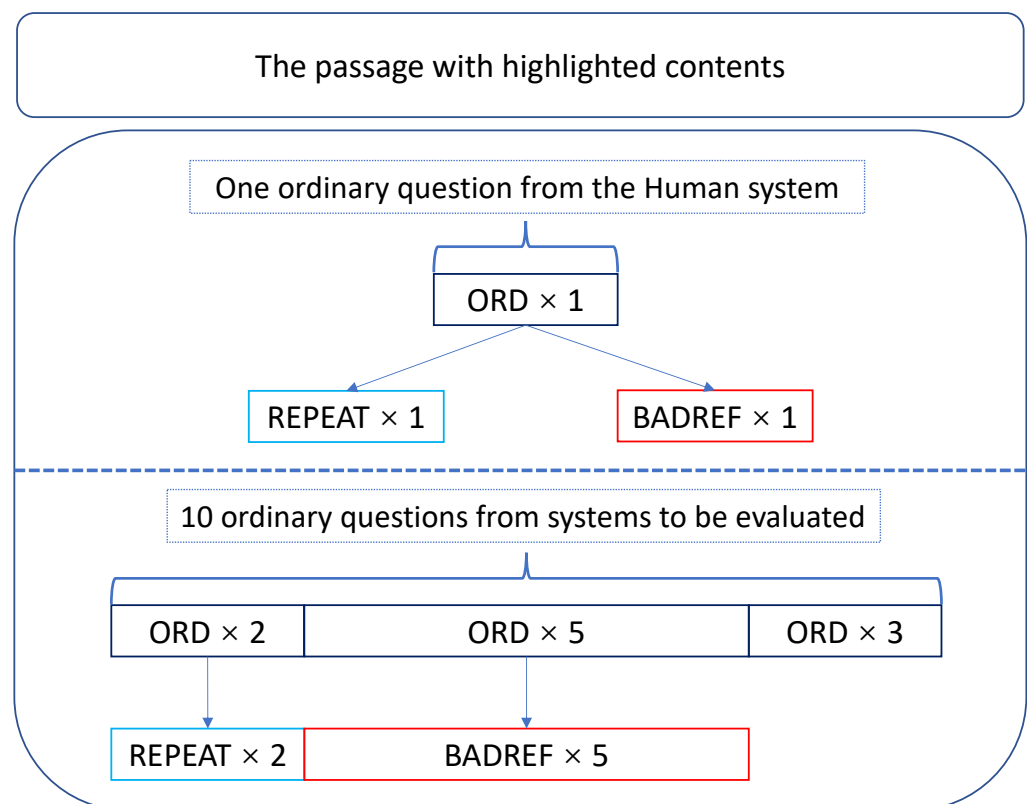


Figure 5. The structure of a single HIT in the QG evaluation experiment, where ORD, REPEAT and BADREF respectively represent ordinary, repeat and bad reference questions.

Each rating item is a question to be rated, together with its passage and answer, where the questions are generated on the HotpotQA test set by 11 various systems, including one system called “Human” that can simulate the human performance, and 10 neural-network-based QG systems, the details of which will be introduced in Section 3.3.

Note that for other tasks involving crowd-sourced human evaluation, a single HIT is made up of 100 items to rate [52]. However, HITs with similar size are inappropriate in this case as a passage containing several sentences should be provided for workers, and a 100-item HIT means a highly oversized workload for an individual. The reading quantity in a single HIT is one of concern as our preliminary experiment shows that a HIT with too many contents to read can significantly decrease the workers’ efficiency. Therefore, we think 20 items in each HIT is more reasonable.

4.3. Experiment Results

In this section, we design an experiment to investigate our proposed human evaluation method, and we report the details of experiments, such as the pass rate of workers and the cost of deploying experiments. We also report the human score of QG systems at the system-level based on the collected data, and deploy a self-replication experiment to inquire into the reliability of this proposed human evaluation method.

4.3.1. Workers and HITs

Two runs of experiments are deployed on the AMT platform, where the second run is designed to serve as a self-replication experiment to ensure the reliability of experimental findings. We then compute the correlation between the human scores of two runs at the system-level to examine the consistency of our method, which will be introduced in Section 4.3.4. The HITs in the two experimental runs are randomly sampled from a HIT pool, which is generated as the outputs from the aforementioned QG systems. Table 4 provides statistical information with regard to the data of workers and HITs collected from our human evaluation experiments.

Table 4. Statistical information of the collected experiment data. (a) The numbers of both workers and HITs before and after the quality-controlling mechanism as well as their pass rates for two runs. (b) The average elapsed time per HIT needed to be completed in minutes, and the average number of HITs that a worker is assigned.

| (a) | | | | | | |
|------------|--------------------------------------|--------|-----------|------------------------------|--------|-----------|
| Experiment | Worker | | | HIT | | |
| | Passed | Total | Pass Rate | Passed | Total | Pass Rate |
| Run1 | 123 | 356 | 34.55% | 334 | 786 | 42.49% |
| Run2 | 105 | 283 | 37.10% | 282 | 598 | 47.16% |
| (b) | | | | | | |
| Experiment | Elapsed Time (per HIT in minutes) | | | Assigned HIT (per Worker) | | |
| | Passed | Failed | Total | Passed | Failed | Total |
| Run1 | 33.24 | 26.93 | 29.61 | 2.72 | 1.94 | 2.21 |
| Run2 | 38.68 | 25.79 | 31.87 | 2.69 | 1.78 | 2.11 |

Table 4a shows the numbers of human raters who participate in the QG evaluation experiment on the AMT platform, who passed the quality control and their pass rate for two distinct runs. The quality control method is as described in Section 4.2. The number of HITs before and after quality control, as well as the pass rate are also reported. For the first run, we collected 334 passed HITs resulting in a total of 18,704 valid ratings. Specifically, a non-human system on average received 1603 ratings and the human system received 2672 ratings, which is a sufficient sample size since it exceeds the minimum

acceptable number (approximately 385) according to the related research of statistical power in MT [54].

Table 4b shows the average duration of a HIT and how many HITs a worker takes on the average according to the influence of the quality control method for both runs. Human raters whose HITs pass the quality control threshold usually spend a longer time completing a HIT than raters of failed HITs.

4.3.2. Cost of the Experiment

Similar to other crowd-sourcing human experiments on the AMT platform [52,53], a worker who passed the quality control was paid 0.99 USD per completed HIT. This entire experiment cost less than 700 USD in total. For research using our proposed evaluation method in the future, the total cost should be approximately half of this since we ran the experiment an additional time to investigate reliability, which generally is not required. The resulting correlation between the system scores of the two separate data collection runs was $r = 0.955$, sufficient to ensure reliability of results. Failed workers were often still paid for their time, where they could claim to have made an honest attempt at the HIT. Only obvious attempts to game the HITs are rejected. In general, according to the cost of our first data collection run, assessing a QG system with nearly 1600 valid ratings in fact costed about 30 USD (total cost 334 USD \div 11 models \approx 30.4 USD). However, the experimental cost in future research may vary, depending on the sample size of collected data.

4.3.3. Human Scores

Human raters may have different scoring strategies, for example, some strict raters tend to give a lower score to the same question compared with other raters. Therefore, we use the average standardized (z) scores instead of the original score, in order to iron out differences resulting from different strategies. Equation (13) is the computation of the average standardized scores for each evaluation criterion and the overall score of a QG system:

$$\begin{aligned} z_q^c &= \frac{r_q^w - \mu_w}{\sigma_w} \\ z^c &= \frac{1}{|Q|} \sum_{q \in Q} z_q^c \\ z &= \frac{1}{|C|} \sum_{c \in C} z^c \end{aligned} \quad (13)$$

where the standardized score z_q^c on the criterion c of a system-generated question q is computed by its raw score r_q^c and the mean μ_w and the standard deviation σ_w of its rater w , z^c is the system-level standardized score on the criterion c of a QG system, Q is the set consisting of all rated questions (q) belonging to the QG system, and the overall average standardized scores z is computed by averaging the z^c of all criteria (C).

Table 5 shows the standardized human scores of all systems based on the ratings from all passed workers in the first run as well as the sample size N , where overall is the arithmetic mean of the scores of understandability, relevancy, answerability and appropriateness. A highlighted value indicates the system in the row outperforms every other system excluding the human Human question for that rating criterion. For the calculation of standardized z scores, the scores of bad references are not included, and for repeat questions the mean score of both evaluations for that question are combined into the final score.

As described in Table 5, the Human system receives the best z scores among all evaluation aspects, which is as expected since it consists of human-generated questions. For all QG systems excluding Human, BART_{large} outperforms all other systems overall, and individually for understandability, relevancy and appropriateness. We also find that BART_{base} somehow performs better than BART_{large} at the answerability criterion. This is

interesting as the performance of a model should generally increase if it is trained on a larger corpus. We think this implies that training models on a larger scale may potentially reduce the ability to generate high quality questions in terms of some aspects, namely answerability in this case. This is probably because a larger corpus may contain more noise which can negatively influence some aspects of a model, and it is worth investigating in future work.

Table 5. Human evaluation standardized z scores of overall and all rating criteria in the first run, where a bold value indicates the system receives the highest score among systems except the Human system, and N indicates the number of evaluated questions of a system; systems (described in Section 3.4) are sorted by the overall score.

| System | N | Overall | Understandability | Relevancy | Answerability | Appropriateness |
|--------------------------|-----|--------------|-------------------|--------------|---------------|-----------------|
| Human | 668 | 0.322 | 0.164 | 0.262 | 0.435 | 0.429 |
| BART _{large} | 400 | 0.308 | 0.155 | 0.255 | 0.420 | 0.403 |
| BART _{base} | 401 | 0.290 | 0.135 | 0.234 | 0.430 | 0.360 |
| T5 _{base} | 395 | 0.226 | 0.051 | 0.241 | 0.395 | 0.217 |
| RNN | 395 | 0.147 | −0.050 | 0.128 | 0.222 | 0.289 |
| Seq2Seq | 404 | 0.120 | −0.030 | 0.022 | 0.180 | 0.309 |
| T5 _{small} | 405 | 0.117 | −0.108 | 0.106 | 0.260 | 0.210 |
| Baseline _{plus} | 408 | 0.076 | −0.133 | 0.076 | 0.196 | 0.165 |
| Seq2Seq* | 396 | 0.053 | −0.055 | −0.039 | 0.088 | 0.217 |
| Baseline | 396 | −0.008 | −0.186 | −0.032 | 0.155 | 0.032 |
| GPT-2 | 408 | −0.052 | −0.202 | −0.126 | 0.050 | 0.068 |

4.3.4. System Consistency

To assess the reliability of the proposed human evaluation method, two distinct runs of the experiment are deployed with different human raters and HITs on the AMT platform. We think a robust evaluation method should be able to have a high correlation between the results of two independent experiment runs.

Table 6 shows the human evaluation results on the second run of our experiment, where the systems follows the order in the first run. We additionally compute the correlation coefficients between the standardized z scores of both runs as shown in Table 7, where r , ρ and τ represent Pearson, Spearman and Kendall's tau correlation, respectively. We observe that the overall scores of two distinct experimental runs can reach $r = 0.955$, while Person correlation of other evaluation criteria ranges from 0.865 (Relevancy) to 0.957 (Answerability). We believe such high correlation values are sufficient to indicate the robustness and reliability of this proposed human evaluation method.

Table 6. Human evaluation standardized z scores of overall and all rating criteria in the second run, where these systems follows the order in Table 5, and N indicates the number of evaluated questions of a system.

| System | N | Overall | Understandability | Relevancy | Answerability | Appropriateness |
|-----------------------|-----|---------|-------------------|-----------|---------------|-----------------|
| Human | 564 | 0.316 | 0.188 | 0.279 | 0.386 | 0.410 |
| BART _{large} | 342 | 0.299 | 0.180 | 0.277 | 0.380 | 0.359 |
| BART _{base} | 338 | 0.306 | 0.181 | 0.299 | 0.397 | 0.347 |

Table 6. Cont.

| System | N | Overall | Understandability | Relevancy | Answerability | Appropriateness |
|--------------------------|-----|---------|-------------------|-----------|---------------|-----------------|
| T5 _{base} | 329 | 0.294 | 0.158 | 0.298 | 0.396 | 0.326 |
| RNN | 342 | 0.060 | −0.040 | −0.008 | 0.072 | 0.217 |
| Seq2Seq | 332 | 0.086 | −0.053 | 0.064 | 0.115 | 0.217 |
| T5 _{small} | 340 | 0.157 | −0.012 | 0.166 | 0.248 | 0.224 |
| Baseline _{plus} | 341 | 0.069 | −0.094 | 0.081 | 0.134 | 0.157 |
| Seq2Seq* | 348 | 0.083 | −0.014 | 0.077 | 0.104 | 0.163 |
| Baseline | 329 | −0.025 | −0.200 | −0.023 | 0.042 | 0.083 |
| GPT-2 | 343 | −0.047 | −0.122 | 0.000 | −0.036 | −0.031 |

Table 7. The Pearson (r), Spearman (ρ) and Kendall's tau (τ) correlations between the standardized z scores of two runs of the experiment, including overall and four evaluation criteria.

| | Overall | Understandability | Relevancy | Answerability | Appropriateness |
|--------|---------|-------------------|-----------|---------------|-----------------|
| r | 0.955 | 0.953 | 0.865 | 0.957 | 0.884 |
| ρ | 0.882 | 0.891 | 0.718 | 0.882 | 0.845 |
| τ | 0.745 | 0.709 | 0.527 | 0.745 | 0.709 |

5. Conclusions and Future Work

In this paper, we propose a new automatic evaluation metric—QAScore, and a new crowd-sourcing human evaluation method for the task of question generation. Compared with other metrics, our metric can evaluate a question only according to its relevant passage and answer without the reference. QAScore utilizes the pretrained language model RoBERTa, and it evaluated a system-generated question by computing the cross entropy regarding the probability that RoBERTa can properly predict the masked words in the answer.

We additionally propose a new crowd-sourced human evaluation method for the task of question generation. Each candidate question is evaluated on four various aspects: understandability, relevancy, answerability and appropriateness. To investigate the reliability of our method, we deployed a self-replication experiment whereby the correlation between the results from two independent runs is shown as high as $r = 0.955$. We also provide a method of filtering out unreliable data from crowd-sourced workers. We introduce the structure of a HIT, the dataset we used and the involved QG systems to encourage the community to repeat our experiment.

According to the results of our human evaluation experiment, we further investigate how performances of QAScore and other metrics. Results show that QAScore achieves the highest correlation with human judgements, which means QAScore can outperform existing QG evaluation metrics.

In conclusion, we propose an unsupervised reference-free automatic metric which correlates better with human judgements compared with other automatic metrics. In addition, we propose a crowd-sourced evaluation method for the question generation task which is highly robust and effective and it can be deployed within a limited budget of time and resources.

In the future therefore, we would like to improve QAScore to achieve fine-grained QG evaluation. Currently, QAScore can only produce an overall score given a question, and we expect to propose an approach to improving QAScore enable it to evaluate a question in different aspects. In addition, we would like to apply QAScore in the evaluation of other language generation tasks, such as dialogue systems or text summarization.

Author Contributions: Conceptualization, Y.G.; data curation, T.J.; funding acquisition, Y.G.; methodology, T.J. and Y.G.; software, T.J. and L.Z.; supervision, Y.G. and G.J.; visualization, C.L.; writing—original draft, T.J. and C.L.; writing—review & editing, T.J., C.L., G.J., L.Z. and Y.G. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by Science Foundation Ireland in the ADAPT Centre for Digital Content Technology (www.adaptcentre.ie, accessed on 23 August 2022) at Trinity College Dublin and Dublin City University funded under the SFI Research Centres Programme (Grants 13/RC/2106_P2; 13/RC/2106) co-funded under the European Regional Development Fund, by Science Foundation Ireland through the SFI Centre for Research Training in Machine Learning (18/CRT/6183).

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We would like to thank the anonymous crowd-sourced raters for their work.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Du, X.; Shao, J.; Cardie, C. 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, BC, Canada, 2017; pp. 1342–1352. [\[CrossRef\]](#)
2. Xie, Y.; Pan, L.; Wang, D.; Kan, M.Y.; Feng, Y. Exploring Question-Specific Rewards for Generating Deep Questions. In *Proceedings of the 28th International Conference on Computational Linguistics*; International Committee on Computational Linguistics: Barcelona, Spain, 2020; pp. 2534–2546. [\[CrossRef\]](#)
3. Pan, L.; Xie, Y.; Feng, Y.; Chua, T.S.; Kan, M.Y. Semantic Graphs for Generating Deep Questions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*; Association for Computational Linguistics: Barcelona, Spain, 2020; pp. 1463–1475. [\[CrossRef\]](#)
4. Puri, R.; Spring, R.; Shoenybi, M.; Patwary, M.; Catanzaro, B. Training Question Answering Models From Synthetic Data. In *Proceedings of 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*; Association for Computational Linguistics: Barcelona, Spain, 2020; pp. 5811–5826. [\[CrossRef\]](#)
5. Lyu, C.; Shang, L.; Graham, Y.; Foster, J.; Jiang, X.; Liu, Q. Improving Unsupervised Question Answering via Summarization-Informed Question Generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*; Association for Computational Linguistics: Punta Cana, Dominican Republic, 2021; pp. 4134–4148.
6. Chen, Y.; Wu, L.; Zaki, M.J. Reinforcement Learning Based Graph-to-Sequence Model for Natural Question Generation. In *Proceedings of the 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, 26–30 April 2020*.
7. Li, J.; Qu, K.; Yan, J.; Zhou, L.; Cheng, L. TEBC-Net: An Effective Relation Extraction Approach for Simple Question Answering over Knowledge Graphs. In *Proceedings of the Knowledge Science, Engineering and Management*; Qiu, H., Zhang, C., Fei, Z., Qiu, M., Kung, S.Y., Eds.; Springer International Publishing: Cham, Switzerland, 2021; pp. 154–165.
8. Kim, Y.; Lee, H.; Shin, J.; Jung, K. Improving Neural Question Generation Using Answer Separation. *Proc. AAAI Conf. Artif. Intell.* **2019**, *33*, 6602–6609. [\[CrossRef\]](#)
9. Wang, L.; Xu, Z.; Lin, Z.; Zheng, H.; Shen, Y. Answer-driven Deep Question Generation based on Reinforcement Learning. In *Proceedings of the 28th International Conference on Computational Linguistics*; International Committee on Computational Linguistics: Barcelona, Spain, 2020; pp. 5159–5170. [\[CrossRef\]](#)
10. Cho, W.S.; Zhang, Y.; Rao, S.; Celikyilmaz, A.; Xiong, C.; Gao, J.; Wang, M.; Dolan, B. Contrastive Multi-document Question Generation. In *Proceedings of 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*; Association for Computational Linguistics: Barcelona, Spain, 2021; pp. 12–30.
11. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.J. 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, PA, USA, 2002; pp. 311–318. [\[CrossRef\]](#)
12. Nema, P.; Khapra, M.M. 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, 2018; pp. 3950–3959. [\[CrossRef\]](#)
13. Sellam, T.; Das, D.; Parikh, A. BLEURT: Learning Robust Metrics for Text Generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*; Association for Computational Linguistics: Brussels, Belgium, 2020; pp. 7881–7892. [\[CrossRef\]](#)
14. Zhang, T.; Kishore, V.; Wu, F.; Weinberger, K.Q.; Artzi, Y. BERTScore: Evaluating Text Generation with BERT. In *Proceedings of the International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, 26–30 April 2020*.

15. Reiter, E. A Structured Review of the Validity of BLEU. *Comput. Linguist.* **2018**, *44*, 393–401. [\[CrossRef\]](#)
16. Graham, Y. Re-evaluating Automatic Summarization with BLEU and 192 Shades of ROUGE. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*; Association for Computational Linguistics: Lisbon, Portugal, 2015; pp. 128–137. [\[CrossRef\]](#)
17. Graham, Y.; Liu, Q. Achieving accurate conclusions in evaluation of automatic machine translation metrics. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, San Diego, CA, USA, 12–17 June 2016; pp. 1–10.
18. Ji, T.; Graham, Y.; Jones, G.J.; Lyu, C.; Liu, Q. Achieving Reliable Human Assessment of Open-Domain Dialogue Systems. *arXiv* **2022**, arXiv:2203.05899.
19. Ji, T.; Lyu, C.; Cao, Z.; Cheng, P. Multi-Hop Question Generation Using Hierarchical Encoding-Decoding and Context Switch Mechanism. *Entropy* **2021**, *23*, 1449. [\[CrossRef\]](#) [\[PubMed\]](#)
20. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR* **2019**, *abs/1907.11692*. Available online: <http://xxx.lanl.gov/abs/1907.11692> (accessed on 20 October 2022).
21. Chen, D.; Fisch, A.; Weston, J.; Bordes, A. Reading Wikipedia to Answer Open-Domain Questions. In *Proceedings of the Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vancouver, BC, Canada, 30 July–4 August 2017; pp. 1870–1879.
22. Zhu, F.; Lei, W.; Wang, C.; Zheng, J.; Poria, S.; Chua, T.S. Retrieving and reading: A comprehensive survey on open-domain question answering. *arXiv* **2021**, arXiv:2101.00774.
23. Rajpurkar, P.; Zhang, J.; Lopyrev, K.; Liang, P. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, TX, USA, 1–4 November 2016; pp. 2383–2392. [\[CrossRef\]](#)
24. Saha, A.; Aralikkatte, R.; Khapra, M.M.; Sankaranarayanan, K. DuoRC: Towards Complex Language Understanding with Paraphrased Reading Comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*; Association for Computational Linguistics: Melbourne, Australia, 2018; pp. 1683–1693. [\[CrossRef\]](#)
25. Kočiský, T.; Schwarz, J.; Blunsom, P.; Dyer, C.; Hermann, K.M.; Melis, G.; Grefenstette, E. The NarrativeQA Reading Comprehension Challenge. *Trans. Assoc. Comput. Linguist.* **2018**, *6*, 317–328. [\[CrossRef\]](#)
26. Xu, Y.; Wang, D.; Yu, M.; Ritchie, D.; Yao, B.; Wu, T.; Zhang, Z.; Li, T.; Bradford, N.; Sun, B.; et al. Fantastic Questions and Where to Find Them: FairytaleQA—An Authentic Dataset for Narrative Comprehension. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*; Association for Computational Linguistics: Dublin, Ireland, 2022; pp. 447–460. [\[CrossRef\]](#)
27. Trischler, A.; Wang, T.; Yuan, X.; Harris, J.; Sordoni, A.; Bachman, P.; Suleman, K. NewsQA: A Machine Comprehension Dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*; Association for Computational Linguistics: Vancouver, BC, Canada, 2017; pp. 191–200. [\[CrossRef\]](#)
28. Lyu, C.; Foster, J.; Graham, Y. Extending the Scope of Out-of-Domain: Examining QA models in multiple subdomains. In *Proceedings of the Third Workshop on Insights from Negative Results in NLP*; Association for Computational Linguistics: Dublin, Ireland, 2022; pp. 24–37. [\[CrossRef\]](#)
29. Lewis, P.; Wu, Y.; Liu, L.; Minervini, P.; Küttler, H.; Piktus, A.; Stenetorp, P.; Riedel, S. PAQ: 65 Million Probably-Asked Questions and What You Can Do with Them. *arXiv* **2021**, arXiv:cs.CL/2102.07033.
30. Zhang, Z.; Zhao, H.; Wang, R. Machine Reading Comprehension: The Role of Contextualized Language Models and Beyond. *arXiv* **2020**, arXiv:cs.CL/2005.06249.
31. Pan, L.; Lei, W.; Chua, T.; Kan, M. Recent Advances in Neural Question Generation. *CoRR* **2019**, *abs/1905.08949*. Available online: <http://xxx.lanl.gov/abs/1905.08949> (accessed on 20 October 2022).
32. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to Sequence Learning with Neural Networks. In *Proceedings of the Advances in Neural Information Processing Systems*; Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., Weinberger, K.Q., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2014; Volume 27.
33. Wu, Y.; Schuster, M.; Chen, Z.; Le, Q.V.; Norouzi, M.; Macherey, W.; Krikun, M.; Cao, Y.; Gao, Q.; Macherey, K.; et al. Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *CoRR* **2016**, *abs/1609.08144*. Available online: <http://xxx.lanl.gov/abs/1609.08144> (accessed on 20 October 2022).
34. Lin, C.Y. ROUGE: A Package for Automatic Evaluation of Summaries. In *Proceedings of the Text Summarization Branches Out*; Association for Computational Linguistics: Barcelona, Spain, 2004; pp. 74–81.
35. Banerjee, S.; Lavie, A. 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, MI, USA, 2005; pp. 65–72.

36. Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.t.; Rocktäschel, T.; et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Proceedings of the Advances in Neural Information Processing Systems*; Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H., Eds.; Curran Associates, Inc.: Ann Arbor, MI, USA, 2020; Volume 33, pp. 9459–9474.
37. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*; Association for Computational Linguistics: Minneapolis, MI, USA, 2019; pp. 4171–4186. [\[CrossRef\]](#)
38. Yuan, X.; Wang, T.; Gulcehre, C.; Sordoni, A.; Bachman, P.; Zhang, S.; Subramanian, S.; Trischler, A. Machine Comprehension by Text-to-Text Neural Question Generation. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*; Association for Computational Linguistics: Vancouver, BC, Canada, 2017; pp. 15–25. [\[CrossRef\]](#)
39. Jia, X.; Zhou, W.; Sun, X.; Wu, Y. EQG-RACE: Examination-Type Question Generation. In *Proceedings of the AAAI, Palo Alto, CA, USA, 2–9 February 2021*.
40. Ren, S.; Zhu, K.Q. Knowledge-Driven Distractor Generation for Cloze-style Multiple Choice Questions. *CoRR* **2020**, *abs/2004.09853*. Available online: <http://xxx.lanl.gov/abs/2004.09853> (accessed on 20 October 2022).
41. Liu, B.; Wei, H.; Niu, D.; Chen, H.; He, Y. Asking Questions the Human Way: Scalable Question-Answer Generation from Text Corpus. In *Proceedings of the Web Conference 2020*, New York, NY, USA, 20–24 April 2020; pp. 2032–2043. [\[CrossRef\]](#)
42. Ma, X.; Zhu, Q.; Zhou, Y.; Li, X. Improving Question Generation with Sentence-Level Semantic Matching and Answer Position Inferring. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 8464–8471. [\[CrossRef\]](#)
43. Narayan, S.; Simões, G.; Ma, J.; Craighead, H.; McDonald, R.T. QURIOUS: Question Generation Pretraining for Text Generation. *CoRR* **2020**, *abs/2004.11026*. Available online: <http://xxx.lanl.gov/abs/2004.11026> (accessed on 20 October 2022).
44. Zhou, S.; Zhang, Y. DATLMedQA: A Data Augmentation and Transfer Learning Based Solution for Medical Question Answering. *Appl. Sci.* **2021**, *11*, 11251. [\[CrossRef\]](#)
45. Shin, T.; Razeghi, Y.; Logan IV, R.L.; Wallace, E.; Singh, S. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 4222–4235. [\[CrossRef\]](#)
46. Lyu, C. Knowledge and Pre-Trained Language Models Inside and Out: A Deep-Dive into Datasets and External Knowledge 2022. Available online: https://scholar.google.co.jp/scholar?hl=zh-TW&as_sdt=0%2C5&q=Knowledge+and+Pre-trained+Language+Models+Inside+and+Out%3A+A+deep-dive+++into+datasets+and+external+knowledge&btnG= (accessed on 20 October 2022).
47. Yang, Z.; Qi, P.; Zhang, S.; Bengio, Y.; Cohen, W.W.; Salakhutdinov, R.; Manning, C.D. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Brussels, Belgium, 31 October–4 November 2018.
48. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.* **2020**, *21*, 1–67.
49. Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; Zettlemoyer, L. 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: Stroudsburg, PA, USA, 2020; pp. 7871–7880. [\[CrossRef\]](#)
50. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language Models are Unsupervised Multitask Learners. *OpenAI Blog* **2019**, *1*, 9.
51. Cho, K.; van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. 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)*, Doha, Qatar, 25–29 October 2014; pp. 1724–1734.
52. Ji, T.; Graham, Y.; Jones, G.J. Contrasting Human Opinion of Non-Factoid Question Answering with Automatic Evaluation. In *Proceedings of the 2020 Conference on Human Information Interaction and Retrieval*; Association for Computing Machinery: New York, NY, USA, 2020; pp. 348–352.
53. Graham, Y.; Baldwin, T.; Moffat, A.; Zobel, J. Can machine translation systems be evaluated by the crowd alone. *Nat. Lang. Eng.* **2017**, *23*, 3–30. [\[CrossRef\]](#)
54. Graham, Y.; Haddow, B.; Koehn, P. Statistical Power and Translationese in Machine Translation Evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 72–81. [\[CrossRef\]](#)