QG-Net: A Data-Driven Question Generation Model for Educational Content

Zichao Wang
Rice University
zichao.wang@rice.edu

Andrew S. Lan
Princeton University
andrew.lan@princeton.edu

Weili Nie
Rice University
weili.nie@rice.edu

Andrew E. Waters
OpenStax
aew2@rice.edu

Phillip J. Grimaldi
OpenStax
phillip.grimaldi@rice.edu

Richard G. Baraniuk
Rice University
richb@rice.edu

ABSTRACT

The ever growing amount of educational content renders it increasingly difficult to manually generate sufficient practice or quiz questions to accompany it. This paper introduces QG-Net, a recurrent neural network-based model specifically designed for automatically generating quiz questions from educational content such as textbooks. QG-Net, when trained on a publicly available, general-purpose question/answer dataset and *without* further fine-tuning, is capable of generating high quality questions from textbooks, where the content is significantly different from the training data. Indeed, QG-Net outperforms state-of-the-art neural network-based and rules-based systems for question generation, both when evaluated using standard benchmark datasets and when using human evaluators. QG-Net also scales favorably to applications with large amounts of educational content, since its performance improves with the amount of training data.

Context: A healthy person in a family in which some members suffer from a recessive genetic disorder may want to know if he or she has the disease-causing gene and what risk exists of passing the disorder on to his or her offspring. Of course, doing **a test cross in humans** is unethical and impractical. Instead, geneticists use **pedigree analysis** to study the inheritance pattern of human genetic diseases.

QG-Net: What is unethical in humans?
QG-Net: What do geneticists use to study the inheritance pattern of human genetic diseases?

Table 1. An example of using QG-Net for question generation. In this example, the context is a paragraph from OpenStax's Biology textbook [27]. The task is to generate questions that are relevant to the context and answer, which are the colored and underlined parts of the context.

INTRODUCTION

Quiz questions remain one of the most important pedagogical tools for learning. Indeed, studies conducted over the last several decades in both traditional educational settings such as classrooms, and large-scale, web-based settings such as massive open online courses (MOOCs) have found that providing students with frequent and ample quiz questions leads to better learning outcomes than spending an equal amount of time studying notes or textbooks [6,15,17–19]. Unfortunately, manually producing such quiz questions is time-consuming because of the extensive effort required of human domain experts. This approach does not scale to the current educational landscape, where an unprecedented growth of educational content (e.g., textbooks, blog posts, lecture notes, magazines, research papers) outpaces the production of questions that accompany them. Therefore, there is a pressing need to find ways to automate the question generation process.

In this work, we study the problem of automatic quiz question generation from educational content. We focus on the setting of generating *factual questions* from a *source context*, such as a sentence or a paragraph in a textbook, and a desired *answer*, which is part of the source context. Table 1 provides an example of the question generation setting that we consider in our study. Factual questions are an important part of assessments, because they assess learners' comprehensive storage of declarative or factual knowledge, which is essential for learning [10]. Moreover, they put learners through the process of stimulating recall, leading to improved knowledge retention [13,16,32]. Thus, factual questions are of substantial value in facilitating learning and improving retention of learned material. Extensions that can generate more advanced questions, such as those involving logical induction or inference, are left for future work.

Within the framework of the problem formulation above, we aim at addressing two major challenges in automatic quiz question generation from educational content. The first major challenge is that the generated questions need to be both *fluent* and *relevant* to be useful for educational applications. Fluency requires the model to generate questions that are free of grammatical errors and, ideally, similar to those that human domain experts would generate. This property is crucial to automated question generation since questions that do not

L@S 2018, June 26–28, 2018, London, United Kingdom
ACM 978-1-4503-5886-6/18/06...$15.00
DOI: https://doi.org/10.1145/3231644.3231654

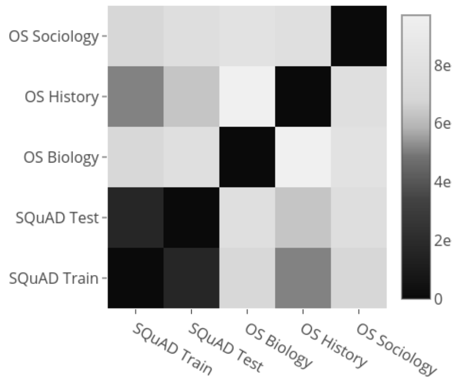

Figure 1. Heatmap showing the similarity between the SQuAD [30], a general purpose dataset, and three OpenStax (OS) textbooks [27] used in our experiments. Darker color indicates higher similarity between the corresponding word frequencies of two text corpora. We see that general-purpose text corpora are very similar, but are significantly different from educational content.

satisfy this requirement can easily mislead and frustrate learners, significantly reducing their pedagogical value. Relevance requires the model to generate questions tailored to specific educational content, i.e., questions that are closely related to the input context and the desired answers. Table 1 shows two different questions that are generated from the same input context but with different answers. This property is also crucial to automated question generation, since it enables the model to generate diverse questions, which can be used as personalized quiz questions that tailor to the knowledge weakness of each individual learner. Generating questions that are both fluent and relevant is difficult, in general, and has been a major topic of ongoing research [11, 40].

The second major challenge is that there is very limited training data in educational applications. Training effective question generation models typically requires both the input contexts and their associated questions. Unfortunately, despite the rapid growth in the amount of educational content, the number of quiz questions that accompany them is insufficient, especially for new content formats like research papers and blog posts. It is thus difficult to train question generation models that achieve fluency and relevance by solely relying on quiz questions in educational settings. However, there exist massive datasets for general-purpose question generation [30]. Although they are not tailored to educational settings, they contain abundant information on how questions should be asked and answered. Therefore, it is interesting to see whether we can reap the power of these massive datasets to improve both question fluency and relevance for question generation in educational settings.

Contributions

We introduce QG-Net, a data-driven question generation model that is particularly suited for the purpose of generating quiz questions on educational content. QG-Net follows the common reader-generator framework [35] but uses several key innovations to ensure the fluency and relevance of the generated questions. First, for the context reader model, we explicitly encode the answer information as an additional input, following [38–40]. This additional information encourages the reader to pay attention to specific parts of the input context, enabling the generator to generate different questions for different answers in the same input context. Second, for the question generator model, we incorporate the pointer network that was originally developed for text summarization [34]. This model design enables the generator to output questions that are able to focus on specific parts of the input text. We show that these aspects of QG-Net significantly improve the *fluency* and *relevance* of the generated questions, using a series of experiments on standard benchmark tasks.

More importantly, we provide the first demonstration of the effectiveness of adapting models trained on general-purpose question generation datasets to educational settings, thus resolving the problem of insufficient training data in these settings. Specifically, we show that QG-Net, trained on SQuAD [30], a massive, publicly available, general-purpose benchmark dataset, can be successfully adapted to OpenStax textbooks [27], generating superior questions than existing RNN-based and rules-based models. The adaptability of QG-Net enables the automatic generation of questions in a broad range of educational domains. We also show that compared to other models, QG-Net scales better with the amount of training data, since it yields higher quality questions when more training data becomes available.

These promising results show that using QG-Net for automated question generation serves as a high-quality and inexpensive alternative to traditional, labor-intensive question generation. QG-Net is ideally suited for large-scale educational settings, where there is an abundance of educational content but a shortage of corresponding quiz questions.

Background and Related Work

Existing question generation models usually represent the input context and output question as *sequences of words*, including punctuation marks. These models consist of a *context reader* that reads the input context and a *question generator* that outputs the question word-by-word. For example, the input context in Table 1 contains 75words, and the output questions from QG-Net contain 7 and 14 words. In other words, these models treat question generation as a *sequence to sequence* learning problem: the reader sequentially processes the input text, from the first word to the last, while the generator outputs one word at a time, until the entire question sentence is generated.

A large body of prior work on question generation relies on extensive feature engineering by parsing the input text and hand-crafting templates or linguistic rules [1, 11, 23, 37]. For example, [11] uses a parser and a set of pre-defined heuristics to generate a large number of questions, and train a ranker with a small, dedicated dataset to select the best one. Due to the meticulously crafted heuristics, these rules-based models perform reasonably well when the language of the input text is well structured. However, when the input context exhibits subtle or complex linguistic features that the hand-crafted

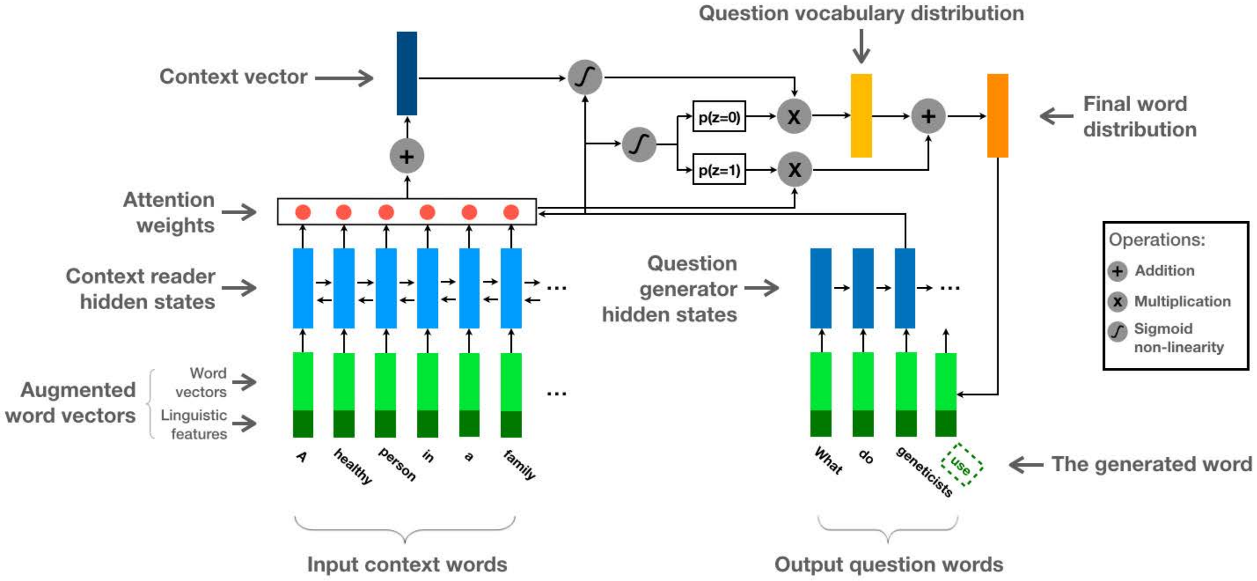

Figure 2. QG-Net model Architecture (best viewed in color).

features overlook, these rules-based models fail to generate satisfactory questions.

Recently, there has been a paradigm shift from rules-based models to data-driven models such as recurrent neural networks (RNNs). Such data-driven models are capable of learning the structures of input and output texts directly from a massive amount of training data [35]. These models are easy to train and have been shown to generate better quality questions than rules-based models [7, 40]. However, almost all RNN-based question generation models are designed to generate questions for machine-related tasks rather than human-related tasks. Specifically, these models lead to improved performance on a series of benchmark tasks, such as machine reading comprehension [7, 39, 40] and machine question answering [8, 36, 38]. As a result, the quality of the questions themselves is often unsatisfactory, since quality is not the primary area of concern in these tasks. Thus, these models are not readily applicable to educational content, since question fluency and relevance are crucial in human-related tasks.

To the best of our knowledge, there exists no prior work on adapting the above question generation models to educational content. One possible reason is that textbooks are written in very different styles than more general-purpose text corpora. For example, Figure 1 shows the difference in word patterns between text from educational content and from a general-purpose question generation dataset, an indication that they have very different content. As a result, there is little evidence to suggest that RNN-based question generation models can be successfully applied to educational content.

THE QG-NET MODEL

We now formulate the question generation problem and detail the QG-Net model. We first lay out the notation that we will use throughout the rest of the paper. Let $C_i = \{c_{ij}\}_{j=1}^{L_i^C}$ be the ith input context sequence (e.g., a paragraph from a textbook) that contains L_i^C words, and $A_i = \{a_{ik}\}_{k=1}^{L_i^A}$ be the length L_i^A answer sequence associated with C_i. Similarly, $Q_i = \{q_{it}\}_{t=1}^{L_i^Q}$ denotes the output question sequence of length L_i^Q that corresponds to the ith input context sequence C_i and answer sequence A_i. In this work, we assume that the answer sequence is a *continuous segment* within the corresponding context; see Table 1 for an example. This assumption unifies the representation of the answer words and other words in the context by using a binary-valued indicator function as an additional feature to indicate whether a word belongs to the answer; an extension to explore the possibility of external answers is left for future work. We make use of *continuous* word representations and represent words as vectors using GloVe [26], an embedding that maps every word into a d-dimensional vector. This vector representation encodes syntactic and semantic relations among words by pre-training on web-scale data [29]. Therefore, the words c_{ij}, a_{ik}, q_{it} in the context, answer, and question are all encoded as d-dimensional vectors. In this work, we set $d = 300$. We drop the index i in the discussion below, unless there is ambiguity.

Problem Formulation

QG-Net generates questions by iteratively sampling question words $q_t \in V^Q$ from the probability distribution

$$P(Q|C,A,\theta) = \prod_{t=1}^{L^Q} P\big(q_t|C,A,\{q_\tau\}_{\tau=1}^{t-1},\theta\big), \qquad (1)$$

where θ denotes the set of parameters.

QG-Net calculates this probability distribution in two steps. First, a *context reader* processes each word c_j in the input context and turns it into a fix-sized representation $h_j \in \mathbb{R}^n$, $j = 1,\ldots,L^C$. Next, a *question generator* generates the question text word-by-word, i.e., one word q_t at each time step $t =$

$1,\ldots,L^Q$, given all context word representations $\left(\{h_j\}_{j=1}^{L^C}\right)$ and all question words in previous time steps $\left(\{q_\tau\}_{\tau=1}^{t-1}\right)$.

Context Reader

The context reader is a bi-directional long short-term memory (bi-LSTM) network [33], an RNNs that is more effective at preserving information from further past in sequential learning tasks. Bi-LSTM processes the input context words sequence in both the forward and backward directions. In each direction, the bi-LSTM iteratively maps i) the jth word vector in the input context and ii) the previous hidden state in this direction into the current (jth) hidden state:

$$\begin{aligned}\overrightarrow{h}_j =& \overrightarrow{\text{bi-LSTM}}(\tilde{c}_j, \overrightarrow{h}_{j-1}),\\ \overleftarrow{h}_j =& \overleftarrow{\text{bi-LSTM}}(\tilde{c}_j, \overleftarrow{h}_{j+1}),\end{aligned}$$

where $\overrightarrow{h_j}$ and $\overleftarrow{h_j}$ are the hidden states corresponding to the forward and the backward directions, respectively. The aggregated hidden state of the jth word in the input context is the concatenation of these two hidden states: $hj = \left[\overrightarrow{h}_j^T, \overleftarrow{h}_j^T\right]^T$. We refer the readers to [14] for details on LSTM networks. $\tilde{c}_j$ is the augmented input word vector corresponding to the jth word, which we detail next.

Encoding the answer into context word vectors

When given different parts of the same input context as answers, QG-Net needs to generate different questions that focus on the relevant contextual information that different answers provide; see Table 1 for an example. To achieve this, QG-Net encodes the kth word in the answer sequence a_k into a 2-dimensional vector as additional features ANS to the word vectors in the input context, which are given by:

$$\text{ANS}_j = \begin{cases}[1,0]^T, & \text{if } a_k = c_{j+k} \; \forall \; k \in [0, L_A],\\ [0,1]^T, & \text{otherwise.}\end{cases} \tag{2}$$

In a similar manner, we encode three additional linguistic features of each word in the input context, including part of speech tag POS, name entity NER, and word case CAS. We use the Stanford natural language processing toolkit [25] to for the POS and NER tags. Each of these features captures additional linguistic information and thus complements the GloVe word vectors. We concatenate these feature encodings with the original GloVe word vectors c_j to form the new word vectors $\tilde{c}_j$ that serve as input to the bi-LSTM context reader:

$$\tilde{c}_j = \left[c_j, \text{ANS}_j^T, \text{POS}_j^T, \text{NER}_j^T, \text{CAS}_j^T\right]^T \in \mathbb{R}^{d+d'}, \tag{3}$$

where d' denotes the aggregated dimension of the additional features. We found that these additional features are a critical piece of side information for QG-Net to generate diverse questions from the *same* input context.

Question Generator

The question generator generates a question word-by-word, from time step $t=1$ to $t=L^Q$, where L^Q is the length of the question that we have defined previously. At each time step t, the question generator generates a question word through the following two internal calculations. First, a uni-directional LSTM network recurrently maps the current (tth) question word into a fix-sized vector, which is the tth hidden state of the network:

$$s_t = \text{LSTM}(y_t, s_{t-1}),$$

where $s_t \in \mathbb{R}^n$ is the hidden state associated with the tth word in the question. Second, a softmax function [9] calculates a probability distribution over all words from a fixed question vocabulary $|V^Q|$:

$$\begin{aligned} e_t &= \sigma(W_s[s_t^T, c_t^{*T}]^T + b_s),\\ P_1(q_t) &= \text{softmax}(W_e e_t^T + b_e),\end{aligned} \tag{4}$$

where e_t is an intermediate variable, σ is the sigmoid function [9], and W_s, b_s, W_e, b_e are model parameters. The vector $c_t^* \in \mathbb{R}^n$ is the context vector, which is the weighted sum of the input hidden states $H = [h_1, ..., h_{L^C}]$:

$$c_t^* = Ha_t,$$

where $a_t \in \mathbb{R}^{L^C}$ is the attention weight vector calculated by the attention mechanism [2] as

$$a_t = \text{softmax}(H^T W_h s_t), \tag{5}$$

where $W_h \in \mathbb{R}^{n \times n}$ is part of the model parameters. From the probability distribution in (4), the generator samples a word $q_t \in V^Q$ at time step t of the generation process, thus deciding the next word in the output question.

Incorporating pointer networks to improve question relevance

Since a good question should be closely related to the context (e.g., by using words directly from the input context), we impose the pointer network [34] on the generator's vocabulary. Specifically, the pointer network calculates the output word probabilities as a mixture of two probabilities, one over the question vocabulary $|V^Q|$ and the other over the input context vocabulary $|V_i^C|$, i.e., the set of unique words in the input context

$$P(q_t) = P(z_t = 0)P(q_t|z_t = 0) + P(z_t = 1)P(q_t|z_t = 1).$$

In the equation above, z_t is a binary-valued variable that switches between generating a word from the question vocabulary and from the input context vocabulary

$$z_t = \begin{cases}0 & \text{if } q_t \in |V^Q|,\\ 1 & \text{if } q_t \in |V_i^C|.\end{cases}$$

The question word q_t is now drawn from the extended vocabulary $V^Q \cup V_i^C$, i.e., the union of the question vocabulary and the vocabulary of the ith input context. Therefore, the probability of generating a word in the question vocabulary is given by (4) as $P(q_t|z_t = 0) = P_1(q_t)$. The probability of generating a word in the input context vocabulary is parameterized by the weight vectors as $P(q_t|z_t = 1) = \sum_{l:q_l=c_l} a_{lt}$. The mixing probability, i.e., $P(z_{it} = 1)$, is calculated from the hidden states s_t as

$$P(z_t = 1) = \sigma(W_z s_t + b_z),$$

where $w_z \in \mathbb{R}^n, b_z \in \mathbb{R}$ are model parameters.

EXPERIMENTS

In this section, we showcase the efficacy of QG-Net through both quantitative and qualitative experiments. We first quantitatively compare QG-Net with several baselines on standard benchmark tasks using a publicly available dataset. We then qualitatively validate QG-Net's adaptivity to educational settings by showing examples of questions that it generates using several textbooks from a wide range of domains.

Quantitative Evaluation

Since educational content does not follow QG-Net's input format (there is no specified context associated with each question), we can only quantitatively evaluate our model by comparing it against baselines using publicly available, general-purpose datasets.

Experiment setup

We train QG-Net on SQuAD, the Stanford Question Answering Dataset [30]. SQuAD contains more than 100k data instances, each of which consists of a short paragraph taken from a Wikipedia article, an answer which is a span of text from the paragraph, and a human generated question based on the paragraph and the answer. We treat the paragraph as the input context to the model and the question as output, thus effectively turning SQuAD into a training dataset for question generation. The dataset explicitly provides us with the indices of the first and last words in the answer. This information makes it straightforward to encode the answer into the corresponding context word vectors.

We truncate each paragraph to only the single sentence that contains the answer and use this sentence as the context during training.[1]

The SQuAD dataset consists of a training set, a validation set, and a test set. They all have the same format. Since the test data is hidden and cannot be accessed, we split the validation set into two halves, and use one half for validation and the other half for test set. During training, we aim to minimize the difference between the generated question and the true question in the training set. We quantify this difference using the negative log likelihood

$$\begin{aligned} L(\theta) &= -\log P(Q|C,A,\theta) \\ &= -\sum_{t=1}^{L_Q} \log P(q_t|C,A,\{q_{i\tau}\}_{\tau=0}^{t-1},\theta). \end{aligned} \tag{6}$$

Since this loss function is differentiable everywhere, we use the standard back-propagation through time (BPTT) with the mini-batch stochastic gradient descent algorithm to learn the model parameters. We employ teacher forcing (i.e., the question generator takes as input the words in the questions in the training set during training), the standard procedure for training LSTMs. During testing, at each time step, the question generator takes its own generated word from the previous time step as input. To generate the best question, we use beam search, a greedy yet effective approximation to exhaustive search, to select the top 25 possible candidate output question sentences. We then choose the one with the lowest negative log likelihood as the final output question. See [9] and references therein for details regarding the training and testing techniques.[2]

Models	**Metrics**		
	BLEU-4	METEOR	ROUGE-L
Over-generate & Rank [12]	0.1120	0.1702	0.2792
LSTM	0.0231	0.0796	0.2703
LSTM + linguistic features	0.0393	0.0972	0.3129
LSTM + attention [7]	0.0658	0.1150	0.3161
LSTM + attention + linguistic features	0.1086	0.1555	0.3988
QG-Net without linguistic features	0.0723	0.1249	0.3368
QG-Net (our full model)	**0.1386**	**0.1838**	**0.4437**

Table 2. A comparison between QG-Net and several baselines on the SQuAD dataset; results show that it outperforms every baseline across all metrics.

Baselines

We compare **QG-Net** with the following baselines: **Over-generate & Rank** [12], a rules-based system that achieves comparable performance to neural network-based models, as reported by [7, 40], **LSTM**, the basic LSTM model, **LSTM + features**, the basic LSTM model with the same linguistic features that we use in our model as additional input, **LSTM + attention** [7], the most recent, state-of-the-art question neural network-based question generation model using the attention mechanism, **LSTM + attention + linguistic features**, the model in [7] augmented with the same linguistic features that we use in our model, and **QG-Net without features**, QG-Net with the linguistic features removed.

Evaluation metrics

Automatically evaluating question generation models is a difficult task, because there are no metrics designed specifically to measure the quality of questions. Therefore, we adopt BLEU [28] and METEOR [21] from machine translation, and ROUGE-L [22] in text summarization as evaluation metrics for question generation, following [7] and [40]. These metrics are calculated by comparing the machine generated question with a human generated reference question from the same input. We refer readers to [21, 22, 28] for details on these metrics.[3] All metric scores take a value in $[0,1]$; higher values indicate higher quality questions. These metrics serve as an initial, inexpensive, large-scale comparison between our model and several other baselines, and can reveal insights into the fluency and relevance of questions generated by each model.

Results and discussion

Table 2 summarizes the comparisons between QG-Net and the baselines on SQuAD test set. We see that QG-Net outperforms all of the baselines on all of the metrics, sometimes significantly so. The results in Table 2 validate the effectiveness of our task-specific modifications to the existing neural

[1] We have also experimented with varying the number of sentences in the input context, and found that the performance is robust to the number of sentences in the input context.

[2] QG-Net code at https://github.com/moonlightlane/QG-Net

[3] The BLEU score counts the co-occurrences of sub-sequences of length N between machine and human generated questions, where $N = 1, 2, 3, 4$ [4]; we report only BLEU-4 for simplicity of exposition.

Context	In 2012, the Economist Intelligence Unit ranked Warsaw as the 32th most liveable city in the world.		
Reference Question	When was Warsaw ranked as the 32nd most liveable city in the world?		
% Training data used	10%	40%	100%
Over-generate & Rank	When did the Economist Intelligence Unit rank Warsaw as the 32nd most liveable city in the world in?	When did the Economist Intelligence Unit rank Warsaw as the 32nd most liveable city in the world in?	When did the Economist Intelligence Unit rank Warsaw as the 32nd most liveable city in the world in?
LSTM + attention + linguistic features	What is the name of the city in the world?	In what year did the CIA conduct Warsaw?	In what year was Warsaw ranked as the fifth most liveable city in the world?
QG-Net	When was the Economist Intelligence ranked?	When was the Economist Intelligence Unit ranked?	When was Warsaw ranked as the 32nd most liveable city in the world?

Table 3. Examples of questions generated by models trained on varying amounts of training data. The input context and reference question are from the SQuAD test set. Answer words are colored and underlined. Over-generate & Rank generates the same questions regardless of the amount of data it is trained on. QG-Net generates higher quality questions than both the rules-based model (Over-generate & Rank) and the neural-network based model (LSTM + attention + linguistic features). The input context comes from the SQuAD test set.

network-based models, namely, adding linguistic features and incorporating vocabulary from the input context. For example, we observe that incorporating additional linguistic features into the input significantly improves the performance of RNN-based models; therefore, these linguistic features contain important side information that is key to generating high-quality questions. We also observe that QG-Net outperforms the LSTM + attention + linguistic features baseline, showing that the ability to copy words into generated questions offers further performance gain.

Scalability with training data

We now show that the performance of QG-Net improves when more training data becomes available. We adopt the same setup from the previous experiment, except that we now vary the amount of training data by training QG-Net on 10%, 40%, 70% and 100% of the SQuAD training set.

Figure 3 plots the ROUGE-L score of our model compared to the Over-generate & Rank and LSTM + attention + linguistic features baselines. These results show that the performance of the neural network-based models increases significantly with more training data, while that of the rules-based system baseline does not. We also observe that QG-Net outperforms the LSTM + attention + linguistic features baseline, and significantly so when the amount of training data is very low, demonstrating its outstanding robustness.

To further illustrate the quality of the questions these models generate using different amounts of training data, we show examples of generated questions in Table 3. The table shows that the rules-based baseline generates the same question, regardless of the amount of training data. While both neural network-based models generate higher quality questions with more training data, the LSTM + attention + linguistic features baseline fails to fully leverage the context (generating "fifth" instead of "32nd" in its output), even with full training data. This observation shows that the LSTM + attention + linguistic features baseline can only improve its mastery of linguistic structure of questions but not its ability to focus on the context as the amount of training data increases. On the contrary, QG-Net quickly learns to focus on answer-relevant information and is able to generate a meaningful question with full training data. This result implies that QG-Net is well-suited to big-data settings, since it can leverage massive training data to improve the quality of the questions it generates.

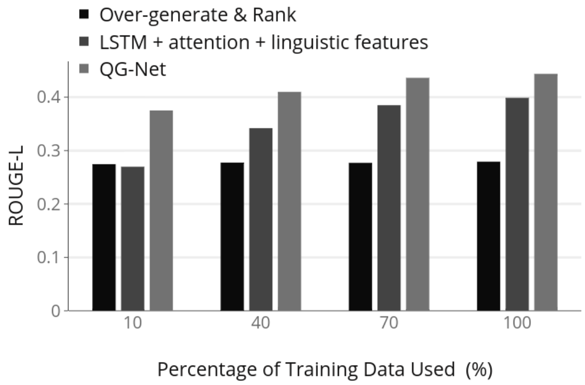

Figure 3. Bar plot comparing the performance of various models trained on varying amounts of training data in terms of the ROUGE-L score on the SQuAD test set. QG-Net outperforms both baselines and thus scales favorably to settings where massive training data is available.

Qualitative Evaluation on OpenStax Textbooks

In this section, we show that transferring the QG-Net model trained on the general-purpose SQuAD dataset to specific subject domains covered by OpenStax (OS) textbooks can lead to fluent and relevant questions. In particular, we take our trained model and apply it to educational content without any further fine-tuning. For this experiment, we choose three OS textbooks: Biology, Sociology, and History. Each of these textbooks covers a different domain, ranging from natural sciences to social sciences and humanities. Each textbook provides roughly four questions at the end of each section but does not provide information on which sentences or paragraphs may help with the answer. As a result, we evaluate the generated questions qualitatively by comparing and contrasting questions generated by QG-Net and the baselines.

Data preparation

Since QG-Net requires both contexts and answers as input, we need to find contexts and the corresponding answer text segments within these contexts. Therefore, we develop a procedure to automatically generate a large set of context-answer pairs. We first split the entire textbooks into chunks

<table>
<tr><td colspan="2"><b>Context (Biology)</b>: On each chromosome, there are thousands of genes that are responsible for determining the genotype and phenotype of the individual. A gene is defined as a sequence of DNA that codes for a functional product. The human haploid genome contains <b>3 billion base pairs</b> and has <b>between 20,000 and 25,000</b> functional genes.</td></tr>
<tr><th><b>3 billion base pairs</b></th><th><b>between 20,000 and 25,000</b></th></tr>
<tr><td><b>QG-Net without linguistic features</b> : How many genes are responsible for determining the genotpye and phenotype?
<b>QG-Net</b> : How many base pairs are on the human genome?</td><td><b>QG-Net without linguistic features</b> : How many genes are responsible for determining the genotpye and phenotype?
<b>QG-Net</b> : How many functional genes are on the human haploid genome?</td></tr>
</table>

Table 4. Example of two generated questions for two different answers with the same input context. The answers are underlined and marked with different colors in the input context. We see that QG-Net generates distinct questions on the *same* input text words for *different* answers. The input context comes from the OpenStax (OS) Biology textbook.

of contexts containing one to five sentences and then use the following heuristics to find feasible answers:

- Word(s) that appear in the index of the textbooks are feasible answers. Textbooks usually contain an index with important terms and phrases in the textbook, which serves as a reasonable source of answers.
- Words that have special NER tags, including "location" (e.g., London), "person" (e.g., John Doe), "date" (e.g., June 2018), "number" (e.g., 1 million), and "organization" (e.g., Association for Computing Machinery). Words or phrases with these tags often contain useful factual information, and can thus be used as possible answers to questions.

Results and discussion

First, we showcase the significant benefit of adding linguistic features to QG-Net. Table 4 compares a question generated by QG-Net with one generated by QG-Net without linguistic features, on a context from OS Biology textbook with two different answers. Results show that QG-Net is able to generate dramatically different questions for *different* answers using the *same* input context. The two answers are only a few words apart and are of the same name entity type "number". QG-Net successfully captures the subtle difference between the relevant information ("base pairs" and "functional genes") and generates a relevant question for each answer. On the contrary, QG-Net without linguistic features is unable to detect this subtle difference and generates the same question even though the answers are different.

Second, we show that questions generated by QG-Net are more fluent and relevant to input contexts and answers than those generated by the baselines. Table 5 compares questions generated by QG-Net and two strong baselines, Over-generate & Rank and LSTM + attention + linguistic features, using 6 different input contexts. In every case, QG-Net generates a question that is more fluent and relevant than those generated by the two baselines.

In terms of fluency, we see that the Over-generate & Rank baseline struggles with complex linguistic structures in the input context (e.g., parallel structure in Contexts 3 and 5, and the presence of semicolon in Contexts 1 and 6), and is unable to generate coherent question sentence in these cases. The other baseline, LSTM + attention + linguistic features, tends to generate grammatically incorrect questions that contain repeated words or phrases (e.g., the word "race" in the question generated from Context 3, and the phrase "white nation" in the question generated from Context 5). On the other hand, QG-Net is able to generate fluent questions even when the input context is lengthy and complex (e.g., Contexts 1 and 5).

In terms of relevance, we see that the Over-generate & Rank baseline often generates questions irrelevant to the answer (e.g., questions generated from Contexts 3, 5 and 6). LSTM + attention + linguistic features baseline generates seemingly relevant questions, but in a number of cases it makes factual errors, generating words or phrases that are not relevant to the input contexts (e.g., the phrase "Kinsey scale" in the question generated from Context 3 and the word "deuterium" in the question generated from Context 4). On the other hand, QG-Net generates questions that are relevant to all the input contexts and answers. We note that there are a few cases where the questions generated by QG-Net and the LSTM + attention + linguistic features baseline are identical (Context 2) or arguably comparable in terms of fluency and relevance (Context 6). Overall, however, questions generated by QG-Net appear to be of higher quality than those generated by the two baselines.

Human Evaluation on OpenStax Textbooks

To further validate the superior quality of the questions generated by QG-Net compared to those generated by baselines, we perform a human evaluation experiment on a number of questions generated from all three OpenStax textbooks.

Experimental setup

For each textbook, we sample 100 input contexts and their associated answers. We then generate one question per context-answer pair using three different models: QG-Net, Over-generate & Rank, and LSTM + attention + linguistic features. During evaluation, the human evaluators are presented with one input context-answer pair at a time, along with questions generated by each model. To avoid possible biases, we randomly shuffle the presentation order of the three questions for each input context. The human evaluators are asked to provide a binary rating (e.g., either true or false) to each of the three questions on two metrics: whether the question is fluent (i.e., coherent and grammatically correct) and whether the question is relevant to the input context-answer pair. As a final metric (dubbed "preference"), the human evaluators are asked to select which question(s) could have been generated by a real human: we allow the evaluators to select more than one question or none of them. This metric reflects the evaluators' subjective judgments of how "human-like" the questions are. We performed the experiments on Amazon Mechanical

Context 1 (Biology): Another example of the link between basic and applied research is the **Human Genome Project**, a study in which each human chromosome was analyzed and mapped to determine the precise sequence of DNA subunits and the exact location of each gene. **Over-generate & Rank**: What is a study? **LSTM + attention + linguistic features**: What is the study between basic and applied research? **QG-Net**: What is the name of the study in which each human chromosome was analyzed?	**Context 2 (Biology)**: Adenosine triphosphate, or **ATP**, is the primary energy currency in cells; ATP stores energy in phosphate ester bonds. **Over-generate & Rank**: What is the primary energy currency in cells; atp stores energy in phosphate ester bonds? **LSTM + attention + linguistic features**: What is the primary energy currency in cell? **QG-Net**: What is the primary energy currency in cells?
Context 3 (Sociology): Weber noted that different groups were affected differently based on education, race, and **gender**, and that peoples reactions to inequality were moderated by class differences and rates of social mobility, as well as by perceptions about the legitimacy of those in power. **Over-generate & Rank**: What was education race, and,? **LSTM + attention + linguistic features**: Along with education, race, race and race, what other groups were affected by the Kinsey scale? **QG-Net**: Along with education and race, what did Weber believe different groups were affected by?	**Context 4 (Sociology)**: In fact, from a structural functionalist perspective, one of the positive contributions of deviance is that it **fosters social change**. **Over-generate & Rank**: What is one of the positive contributions of deviance that it fosters in fact? **LSTM + attention + linguistic features**: What is one of the positive contributions of deuterium? **QG-Net**: What is one of the positive contributions of deviance?
Context 5 (History): The 1830 **Indian Removal Act** and subsequent displacement of the Creek, Choctaw, Chickasaw, Seminole, and Cherokee tribes of the southeast fulfilled the vision of a white nation and became one of the identifying characteristics of the age of Jackson. **Over-generate & Rank**: What did and subsequent displacement of the Creek, Choctaw, Chickasaw, Seminole, and Cherokee tribes of the southeast fulfil the vision of a white nation? **LSTM + attention + linguistic features**: What was the name of the act that caused a white nation to become a white nation? **QG-Net**: What act became one of the identifying characteristics of the age of Jackson?	**Context 6 (History)**: Attendees agreed to a declaration of rights and sentiments based on the **Declaration of Independence**; It declared, we hold these truths to be self-evident: that all men and women are created equal; that they are endowed by their creator with certain inalienable rights; that among these are life, liberty, and the pursuit of happiness. **Over-generate & Rank**: Was it the first of what became annual meetings that have continued to the present day? **LSTM + attention + linguistic features**: What document was given to the declaration of rights and themes? **QG-Net**: Attendees agreed to a declaration of rights and sentiments based on what document?

Table 5. Example of three questions generated by three different models from various input contexts and answers. We see that, in all cases, QG-Net generates questions that are both fluent and relevant, whereas in most cases, the two baselines fail to do so. The input contexts come from the OS Biology, Sociology, and History textbooks.

Turk [3] and collected 4 responses for each generated question. We then calculate, for each question, the statistical mode of the evaluations for each of the three metrics. This procedure enables us to resolve disagreements among raters and results in a single label for each metric.

Results and discussion

Figure 4 summarizes the human evaluation results in 3 separate bar plots for the OS Biology, Sociology, and History textbooks. In all three plots, we count the number of questions in which the majority of raters gave a positive evaluation for each of the three question generation models. In all cases, QG-Net (often significantly) outperforms the two baselines on all three evaluation metrics. We evaluate the statistical significance of these results using a binomial test, and find that the degree to which QG-Net outperforms the two baselines is statistically significant well beyond the $p = 0.05$ level for all evaluation metrics except for fluency and relevance for the History textbook ($p = 0.11$ and $p = 0.5$, respectively). We emphasize that, in the case of evaluating whether the questions could have been generated by a human, QG-Net significantly outperforms the two baselines (beyond the $p = 0.01$ level for all textbooks). Moreover, in the majority of cases (more than 60 out of 100 questions across all textbook subjects), QG-Net generates a question that is deemed “human-like” by human evaluators.

We thus conclude that QG-Net generates questions that are fluent, relevant, and “human-like” more often than existing models. These results imply that questions generated by QG-Net have better applicability in real-world educational settings than those generated by other baselines.

LIMITATIONS OF QG-NET AND FUTURE WORK

In this section, we discuss two major limitations of QG-Net and discuss possible future works to address these limitations.

First, although QG-Net generates significantly better questions than previous models, it is not guaranteed to always generated good questions, since there exist no effective question evaluation metrics that can automatically filter out bad questions. The absence of such metrics means that QG-Net is not yet ready for large-scale automated deployment in real-world educational settings, since human experts are still required to review the generated questions before assigning them to learners. However, experts’ involvement provides an opportunity for developing novel and interactive “human-in-the-loop” systems. Specifically, we can first use QG-Net to generate a large number of quiz questions at low cost. Then, we can leverage

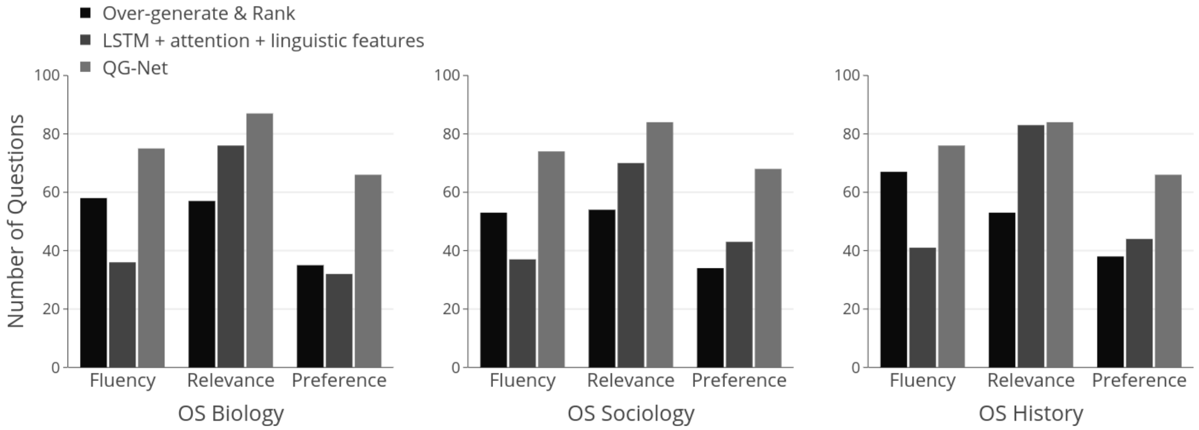

Figure 4. Bar plots comparing the performance of QG-Net with two strong baselines on three human evaluation metrics. Each bar plot shows results on one of the OS textbooks. In the majority of cases, QG-Net generates questions that are more fluent, more relevant to the input context and answer, and are considered as similar to those human would have generated more often than the baselines.

feedback on the quality of the generated questions provided by either human experts or by testing their pedagogical values [20, 24] to further improve QG-Net's capability. These interactive systems have the potential to improve with increasing usage, which is an ideal fit for large-scale educational applications.

Second, QG-Net is only capable of generating factual questions. While factual questions are valuable for learning (see the Introduction), this constraint limits the depth of the questions. Several recent works make use of first order logic that enables model to perform reasoning to some extent [5, 31], but these works do not directly apply to the task of question generation. A combination of first order logic and neural networks thus holds promise for generating more advanced questions.

CONCLUSION

We have introduced QG-Net, an RNN-based question generation model specifically designed for generating quiz questions from educational content such as textbooks. Our model design leverages several recent advances in text summarization and question answering. We have demonstrated the superior performance of QG-Net over several baselines on a standard benchmark dataset. More importantly, we have demonstrated that, after training QG-Net on a general-purpose question generation dataset, we can adapt it to educational content and generate fluent and relevant questions, without further finetuning. These promising results suggest that QG-Net has the potential to automate and scale up the question generation process for educational settings where a large number of quiz and practice questions are needed to accompany abundant educational content.

ACKNOWLEDGEMENTS

This research was supported by the Arthur & Carlyse Ciocca Charitable Foundation, The Laura and John Arnold Foundation, John and Ann Doerr, IBM Research, and NSF grant DRL-1631556. Thanks to Bob Schloss for his insights on the work.

REFERENCES

1. I. Aldabe, M. L. de Lacalle, M. Maritxalar, E. Martinez, and L. Uria. 2006. ArikIturri: An Automatic Question Generator Based on Corpora and NLP Techniques. In *Proc. International Conference on Intelligent Tutoring Systems*. 584–594.
2. D. Bahdanau, K. Cho, and Y. Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proc. International Conference on Learning Representations*.
3. M. Buhrmester, T. Kwang, and Gosling S. D. 2011. Amazon's Mechanical Turk: A New Source of Inexpensive, Yet High-Quality, Data? *Perspectives on Psychological Science* 6, 1 (2011), 3–5.
4. X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollar, and C. L. Zitnick. 2015. Microsoft COCO Captions: Data Collection and Evaluation Server. *arXiv e-prints* (Apr. 2015). `https://arxiv.org/abs/1504.00325`
5. W. W. Cohen, F. Yang, and K. Mazaitis. 2017. TensorLog: Deep Learning Meets Probabilistic DBs. *CoRR* abs/1707.05390 (2017). `http://arxiv.org/abs/1707.05390`
6. P. A. Connor-Greene. 2000. Assessing and Promoting Student Learning: Blurring the Line between Teaching and Testing. *Teaching of Psychology* 27, 2 (2000), 84–88.
7. X. Du, J. Shao, and C. Cardie. 2017. Learning to Ask: Neural Question Generation for Reading Comprehension. In *Proc. Annual Meeting of the Association for Computational Linguistics*. 1342–1352.
8. N. Duan, D. Tang, P. Chen, and M. Zhou. 2017. Question Generation for Question Answering. In *Proc. Conference on Empirical Methods in Natural Language Processing*. 866–874.
9. I. Goodfellow, Y. Bengio, and A. Courville. 2016. *Deep Learning*. MIT Press.
10. R. E. Haskell. 2011. *Transfer of Learning: Cognition, Instruction, and Reasoning*. Academic Press.
11. M. Heilman. 2011. *Automatic Factual Question Generation from Text*. Ph.D. Dissertation.

12. M. Heilman and N. A. Smith. 2010. Good Question! Statistical Ranking for Question Generation. In *Proc. Conference on Human Language Technologies*. 609–617.

13. R. J. Hift. 2014. Should essays and other "open-ended"-type questions retain a place in written summative assessment in clinical medicine? *BMC Medical Education* 14, 1 (Nov. 2014), 249.

14. S. Hochreiter and J. Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (Nov. 1997), 1735–1780.

15. J. Karpicke. 2012. Retrieval-Based Learning: Active Retrieval Promotes Meaningful Learning. *Current Directions in Psychological Science* 21, 3 (May 2012), 157–163.

16. J. D. Karpicke and P. J. Grimaldi. 2012. Retrieval-based learning: A perspective for enhancing meaningful learning. *Educational Psychology Review* 24, 3 (Sep. 2012), 401–418.

17. J. D. Karpicke and H. L. Roediger. 2008. The Critical Importance of Retrieval for Learning. *Science* 319, 5865 (2008), 966–968.

18. K. R. Koedinger, J. Kim, J. Z. Jia, E. A. McLaughlin, and N. L. Bier. 2015. Learning is Not a Spectator Sport: Doing is Better Than Watching for Learning from a MOOC. In *Proc. Conference on Learning at Scale*. 111–120.

19. G. Kovacs. 2016. Effects of In-Video Quizzes on MOOC Lecture Viewing. In *Proc. Conference on Learning at Scale*. 31–40.

20. A. S. Lan and R. G. Baraniuk. 2016. A Contextual Bandits Framework for Personalized Learning Action Selection. In *Proc. Intl. Conf. on Educational Data Mining*. 424–429.

21. A. Lavie and A. Agarwal. 2007. Meteor: An Automatic Metric for MT Evaluation with High Levels of Correlation with Human Judgments. In *Proc. Workshop on Statistical Machine Translation*. 228–231.

22. C. Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Proc. Workshop on Text Summarization Branches Out*. 74–81.

23. M. Liu, R. A. Calvo, and V. Rus. 2010. Automatic Question Generation for Literature Review Writing Support. In *Proc. International Conference on Intelligent Tutoring Systems*. 45–54.

24. I. Manickam, A. S. Lan, and R. G. Baraniuk. 2017. Contextual multi-armed bandit algorithms for personalized learning action selection. In *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing*. 6344–6348.

25. C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proc. Association for Computational Linguistics*. 55–60.

26. T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Proc. Advances in Neural Information Processing Systems*. 3111–3119.

27. OpenStax. 2017. OpenStax Textbooks. (2017). `https://openstax.org/`

28. K. Papineni, S. Roukos, T. Ward, and W. Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proc. Annual Meeting of the Association for Computational Linguistics*. 311–318.

29. J. Pennington, R. Socher, and C. D. Manning. 2014. Glove: Global Vectors for Word Representation. In *Proc. Conference on Empirical Methods in Natural Language Processing*. 1532–1543.

30. P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proc. Conference on Empirical Methods in Natural Language Processing*. 2383–2392.

31. M. Richardson and P. Domingos. 2006. Markov logic networks. *Machine learning* 62, 1-2 (2006), 107–136.

32. D. Rohrer and P. Pashler. 2010. Recent Research on Human Learning Challenges Conventional Instructional Strategies. *Educational Researcher* 39, 5 (Oct. 2010), 406–412.

33. M. Schuster and K. K. Paliwal. 1997. Bidirectional Recurrent Neural Networks. *IEEE Trans. Signal Processing* 45, 11 (Nov. 1997), 2673–2681.

34. A. See, P. J. Liu, and C. D. Manning. 2017. Get To The Point: Summarization with Pointer-Generator Networks. In *Proc. Annual Meeting of the Association for Computational Linguistics*. 1073–1083.

35. I. Sutskever, O. Vinyals, and Q. V. Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Proc. Advances in Neural Information Processing Systems*. 3104–3112.

36. D. Tang, N. Duan, T. Qin, Z. Yan, and M. Zhou. 2017. Question Answering and Question Generation as Dual Tasks. *ArXiv e-prints* (2017). `https://arxiv.org/abs/1706.02027`

37. J. H. Wolfe. 1976. Automatic Question Generation from Text — an Aid to Independent Study. In *Proc. Symposium on Computer Science and Education*. 104–112.

38. Z. Yang, J. Hu, R. Salakhutdinov, and W. Cohen. 2017. Semi-Supervised QA with Generative Domain-Adaptive Nets. In *Proc. Annual Meeting of the Association for Computational Linguistics*. 1040–1050.

39. X. Yuan, T. Wang, C. Gulcehre, A. Sordoni, P. Bachman, S. Subramanian, S. Zhang, and A. Trischler. 2017. Machine Comprehension by Text-to-Text Neural Question Generation. *ArXiv e-prints* (May 2017). `https://arxiv.org/abs/1705.02012`

40. Q. Zhou, N. Yang, F. Wei, C. Tan, H. Bao, and M. Zhou. 2017. Neural Question Generation from Text: A Preliminary Study. *ArXiv e-prints* (2017). `https://arxiv.org/abs/1704.01792`