CliniQG4QA: Generating Diverse Questions for Domain Adaptation of Clinical Question Answering

Xiang Yue[1,*], Xinliang (Frederick) Zhang[1,*], Ziyu Yao[1], Simon Lin[2], and Huan Sun[1]

[1]The Ohio State University
[2]Abigail Wexner Research Institute at Nationwide Children's Hospital
{yue.149, zhang.9975, yao.470, sun.397}@osu.edu
Simon.Lin@nationwidechildrens.org

Abstract

Clinical question answering (QA) aims to automatically answer questions from medical professionals based on clinical texts. Studies show that neural QA models trained on one corpus may not generalize well to new clinical texts from a different institute or a different patient group, where large-scale QA pairs are not readily available for retraining. To address this challenge, we propose a simple yet effective framework, CliniQG4QA, which leverages question generation (QG) to synthesize QA pairs on new clinical contexts and boosts QA models without requiring manual annotations. In order to generate diverse types of questions that are essential for training QA models, we further introduce a seq2seq-based question phrase prediction (QPP) module that can be used together with most existing QG models to diversify their generation. Our comprehensive experiment results show that the QA corpus generated by our framework is helpful to improve QA models on the new contexts (up to 8% absolute gain in terms of Exact Match), and that the QPP module plays a crucial role in achieving the gain.[1]

1 Introduction

Clinical question answering (QA), which aims to automatically answer natural language questions based on clinical texts in Electronic Medical Records (EMR), has been identified as an important task to assist clinical practitioners (Patrick and Li, 2012; Raghavan et al., 2018; Pampari et al., 2018; Fan, 2019; Rawat et al., 2020). Neural QA models in recent years (Chen et al., 2017; Devlin et al., 2019; Rawat et al., 2020) show promising results in this research. However, answering clinical questions still remains challenging in real-world scenarios, because well-trained QA systems may not generalize well to new clinical contexts from a different institute or a different patient group. For example, as pointed out in (Yue et al., 2020), when a clinical QA model that was trained on the emrQA dataset (Pampari et al., 2018) is deployed to answer questions based on MIMIC-III clinical texts (Johnson et al., 2016), its performance drops dramatically by around 30% even on the questions that are similar to those in training, simply because clinical texts of the two datasets are different (e.g., different topics, note structures, writing styles).

One straightforward solution is to annotate QA pairs on new contexts and retrain a QA model. However, manually creating large-scale QA pairs in clinical domain is extremely challenging due to the requirement of tremendous expert effort, data privacy concerns and other ethical issues.

In this work, we study the problem of *constructing clinical QA models on new contexts without human-annotated QA pairs* (which is referred to as domain adaptation). We assume the availability of a large set of QA pairs on *source* contexts, and our goal is to better answer questions on new documents (*target* contexts[2]), where only unlabeled documents are provided.

To this end, we introduce our framework, CliniQG4QA, which leverages question generation (QG), a recent technique of automatically generating questions from given contexts (Du et al., 2017), to synthesize clinical QA pairs on target contexts to facilitate the QA model training (Figure 1). The QG model is built up by reusing the QA pairs on source contexts as training data. To apply QG to target contexts, our framework also includes an answer evidence extractor to extract meaningful text spans, which are worthwhile to ask questions about, from the clinical documents. Intrinsically, our framework is backed by the observation that questions in the clinical domain generally follow similar patterns even across different contexts, and

*The first two authors contributed equally.

[1]Our code is available at: https://github.com/sunlab-osu/CliniQG4QA

[2]We use "new" and "target" contexts interchangeably.

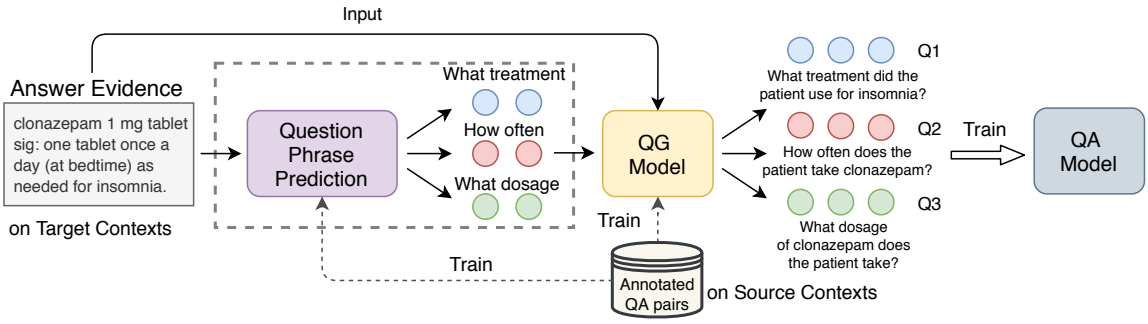

Figure 1: Illustration of our simple yet effective CliniQG4QA framework: A key component is our question phrase prediction (QPP) module, which aims to generate diverse question phrases and can be "plugged-and-played" with most existing QG models to diversify their generation.

clinical QG suffers less from the context shift compared with clinical QA. This allows us to utilize QG models trained on source clinical contexts to boost QA models on different target contexts.

However, our preliminary studies find that many existing QG models often fall short on generating questions that are *diverse* enough to serve as useful training data for clinical QA models. To tackle the problem, we introduce a *question phrase prediction* (QPP) module (Figure 1), which takes an answer evidence as input and sequentially predicts potential question phrases (e.g., "What treatment", "How often") that signify what types of questions humans would likely ask about the answer evidence. By directly forcing a QG model to produce specified question phrases in the beginning of the question generation process (both in training and inference), QPP enables diverse questions to be generated.

We conduct extensive experiments to evaluate CliniQG4QA, using emrQA (Pampari et al., 2018) as the source contexts and MIMIC-III (Johnson et al., 2016) as the target ones. We instantiate our framework with a variety of widely adopted base QG models and base QA models.

By performing comprehensive analyses, we first show that the proposed QPP module can substantially improve both the generation relevance (up to 8% absolute BLEU improvement) and diversity (up to 5% absolute Distinct score gain) of base QG models. It encourages the QG models to generate much more diverse types of questions (e.g., "When" and "Why" questions) to reflect the variations in human-made questions.

More importantly, we systematically demonstrate the strong capability of CliniQG4QA for improving QA performance on new contexts by evaluating it on a set of QA pairs that we ask human experts to annotate on MIMIC-III. When using QA pairs automatically synthesized by our QPP-enhanced QG models as the training corpus, we are able to boost QA models' performance by up to 8% in terms of Exact Match (EM), compared with their counterparts directly trained on the emrQA dataset. To further investigate why QG boosts QA, we provide both quantitative and qualitative analyses, indicating that QA models can benefit from seeing more target contexts as well as more diverse questions generated on them.

2 Related Work

Clinical Question Answering aims to automatically answer user questions based on different sources (Cairns et al., 2011; Patrick and Li, 2012; Roberts and Patra, 2017; Raghavan et al., 2018; Soni et al., 2019). In this study, we focus on clinical reading comprehension task, which aims to extract a text span (a sentence or multiple sentences) as the answer from a patient clinical note given a question (Yue et al., 2020). Though many neural models (Seo et al., 2017; Chen et al., 2017; Devlin et al., 2019; Rawat et al., 2020; Wen et al., 2020) have achieved impressive results on this task, their performance on new clinical contexts, whose data distributions could be different from the ones that these models were trained on, is still far from satisfactory (Yue et al., 2020). Though one can improve the performance by adding more QA pairs on new contexts into training, however, manually creating large-scale QA pairs in the clinical domain often involves tremendous expert effort and data privacy concerns.

Question Generation seeks to automatically generate questions given a sentence or paragraph. Existing QG models (Du et al., 2017; Zhou et al., 2017; Sun et al., 2018; Zhao et al., 2018; Nema et al., 2019; Tuan et al., 2020; Yang et al., 2017; Du and Cardie, 2018; Alberti et al., 2019; Zhang and Bansal, 2019) in the open domain usually adopt

a seq2seq (encoder-decoder) architecture. One of the drawback of such models is that they can only generate one question given one input and fail to generate multiple diverse questions, which we find is crucial to the QA task. Some recent work (Kang et al., 2019; Cho et al., 2019; Liu et al., 2020) explore the diverse QG in the open domain, but they cannot be directly applied to the clinical domain as their models usually require a short answer (e.g., an entity) as input but that information sometimes is not available in the clinical QA dataset (e.g. emrQA (Pampari et al., 2018)), rendering the difficulty of directly deploying their model on the clinical QA.

In the clinical and medical domain, (Shen et al., 2020) and (Soni and Roberts, 2019, 2020) apply Variational Autoencoder (VAE) models to generate or paraphrase medical or clinical questions. However, none of them explore leveraging QG to improve QA performance on new contexts.

Domain Adaptation for Reading Comprehension. Our work is also related to domain adaptation, a technique of generalizing machine learning models to unseen domains (Ben-David et al., 2010). In the field of reading comprehension, most works (Wiese et al., 2017; Chung et al., 2018; Hazen et al., 2019; Cao et al., 2020) typically assume that questions on the target contexts are available, however, manually creating questions in the clinical domain is not trivial.

Our work is most related to (Golub et al., 2017; Wang et al., 2019) in the open domain where only unlabeled documents exist in the target domain. Similar to ours, they leverage a QG model trained on source contexts to generate questions for new contexts. While the QG models used in both work are shown helpful for QA, we empirically observe that they tend to generate questions of the most frequent types but fail to generate *diverse* questions which can be more helpful for training QA models. We address this issue by proposing a simple yet effective question phrase prediction module, which learns to predict a set of question phrases and can be used together with any neural QG models to diversify their generation. Experiments demonstrate that questions generated by our QPP-enhanced QG models are more helpful to boost QA models.

3 Methods

3.1 Overview of Our Framework

We first give an overview of our `CliniQG4QA` framework (Figure 1). `CliniQG4QA` improves clinical QA on new contexts by automatically synthesizing QA pairs for new clinical contexts. To approach this, we first leverage an *answer evidence extractor* to extract meaningful text spans from unlabeled documents, based on which a QG model can be applied to generate questions.

In order to encourage diverse questions, we reformulate the question generation process as two-stage. In the first stage, we propose a *question phrase prediction* module to predict a set of question phrases, which represent the types of questions humans would ask, given an answer evidence. In the second stage, following a specific question phrase predicted by our QPP, a QG model is used to complete the rest of the question.

Therefore, our framework `CliniQG4QA` is able to produce questions of more diverse types. The generated QA pairs by QG models are finally used to train QA models on the new contexts.

3.2 Answer Evidence Extractor (AEE)

When human annotators create questions, they first read a document and then select a text span to ask questions about. To imitate this process, we implement an answer evidence extractor to extract possible text spans from a document. Following (Pampari et al., 2018; Yue et al., 2020), we focus on longer text spans (as answer evidences) instead of short answers (e.g., a single named entity), since longer text spans often contain richer information compared with short ones, which are very important in clinical QA.

More formally, given a document (context) $\mathbf{p} = \{p_1, p_2, ..., p_m\}$, where p_i is the i-th token of the document and m is the total number of tokens, we aim to extract potential evidence sequences. Since the answer evidence is not always a single sentence (sometimes could be multiple sentences), instead of treating it as a sentence selection task, we formulate it as a sequence labeling (or tagging) task. We follow the `BIO` tagging (short for beginning, inside, outside), a commonly used sequence labeling scheme (Ramshaw and Marcus, 1999), to label answer evidences.

Firstly, we adopt the ClinicalBERT model (Alsentzer et al., 2019) to encode the document:

$$\mathbf{U} = \texttt{ClinicalBERT}\{p_1, ..., p_m\}. \quad (1)$$

where $\mathbf{U} \in \mathbb{R}^{m \times d}$, and d is size of the dimension.

Following the same paradigm of the BERT model for the sequence labeling task (Devlin et al., 2019), we use a linear layer on top of the hidden

states output by BERT followed by a softmax function to do the classification:

$$\Pr(a_j|p_i) = \text{softmax}(\mathbf{U} \cdot \mathbf{W} + \mathbf{b}),\ \forall p_i \in \mathbf{p} \quad (2)$$

where a_j is the predicted BIO tag.

After prediction, we develop some heuristic rules (e.g., removing very short extracted evidences, merging evidences that sit close to each other) to further improve the quality of the extracted evidences. All heuristic rules we used are listed in the Appendix A.

3.3 Question Phrase Prediction (QPP)

Existing QG models are often biased to generate limited types of questions. To address this problem, we introduce our question phrase prediction module that can be used to diversify the generation of existing QG models.

Formally, denote $V_l = \{s_1, ..., s_L\}$ as the vocabulary of all available question phrases of length l in the training data and $L = |V_l|$ as its size. V_l can be obtained by collecting the first n-gram words in the questions (more details are in Appendix B.1.2). Given an answer evidence $\mathbf{a}$, the goal of QPP is to map $\mathbf{a} \rightarrow \mathbf{y} = (y_1, ..., y_L) \in \{0, 1\}^L$, where $y_i = 1$ indicates predicting s_i in V_l as a question phrase for the evidence. Instead of treating it as a common multi-label classification problem, we formulate the task as *a sequence prediction* problem and adopt a commonly used seq2seq model with an attention mechanism (Luong et al., 2015) to predict a sequence of question phrases $\mathbf{s} = (s_{j_1}, ..., s_{j_{|\mathbf{s}|}})$ (e.g., "What treatment" (s_{j_1}) $\rightarrow$ "How often" (s_{j_2}) $\rightarrow$ "What dosage" (s_{j_3}), with $|\mathbf{s}| = 3$).

During training, we assume that the set of question phrases is arranged by following a pre-defined order. Such orderings can be obtained with some heuristic methods, e.g., using a descending order based on question phrase frequency in the corpus.

Our QPP can dynamically decide the number of appropriate question phrases for each answer evidence in the inference stage while standard multi-label classification methods need a static value as threshold (e.g., Top-3). Moreover, the dynamic property can increase credibility of diverse generation since the diversifying step is conducted in a case-by-case analysis fashion instead of blindly determining the sampling size (of question phrase set) during inference, which is unfortunately commonly utilized in recent diverse QG works (Cho et al., 2019; Liu et al., 2020).

Algorithm 1 CliniQG4QA training procedure

Input: labeled *source* data $\{(P_S, A_S, Q_S)\}$, unlabeled *target* data $\{P_T\}$

Output: Generated QA pairs $\{(A'_T, Q'_T)\}$ on *target* contexts; An optimized QA model for answering questions on target contexts;

Pretraining Stage

1: Train *Answer Evidence Extractor* based on the *source* data $\{(P_S, A_S)\}$ using Eq. 3
2: Obtain question phrase data Y_S from Q_S and train *Question Phrase Prediction* module on the *source* data $\{(A_S, Y_S)\}$ using Eq. 4.
3: Train a *QPP-enhanced QG* model on the *source* data $\{(A_S, Y_S, Q_S)\}$ using Eq. 5.

Training Stage

4: Use *AEE* to extract potential answer evidences $\{A'_T\}$ on the *target* contexts $\{P_S\}$
5: Use *QPP* to predict potential question phrases set $\{Y'_T\}$ on $\{A'_T\}$
6: Use *QPP-enhanced QG* to generate diverse questions $\{Q'_T\}$ based on $\{(A'_T, Y'_T)\}$
7: Train a *QA* model on synthetic *target* data $\{(P_T, A'_T, Q'_T)\}$

3.4 Training

Algorithm 1 illustrates the pretraining and training procedure of our CliniQG4QA.

During the *pretraining* stage, we first train the answer evidence extractor (AEE), question phrase prediction (QPP) module on the source contexts by minimizing the negative log-likelihood loss:

$$L_{AEE} = -\sum_i \log P(\mathbf{a}|\mathbf{p}; \phi) \quad (3)$$

where ϕ represents all the parameters of the answer evidence extractor. For the supervision signals, we identify all evidence evidences in the source data as ground-truth chunks which are marked using the BIO scheme.

Moving to Question Phrase Prediction (QPP) module, given an answer evidences $\mathbf{a}$, we aim to predict a question phrase sequence $\mathbf{y}$ and minimize:

$$L_{QPP} = -\sum_i \log P(\mathbf{y}|\mathbf{a}; \theta) \quad (4)$$

where θ denotes all the parameters of QPP.

Then we can train any QG model (e.g, NQG (Du et al., 2017)) on source data by minimizing:

$$L_{QG} = -\sum_i \log P(\mathbf{q}|\mathbf{a}, \mathbf{y}; \gamma) \quad (5)$$

where γ denotes all parameters of the QG model.

During the *training* stage, given the unlabeled target clinical documents, we first extract potential answer evidences, based on which QPP can be "plugged" into the QG model to generate diverse questions. Finally, a QA model can be trained on the generated QA pairs of the target documents.

4 Experimental Setup

4.1 Datasets

Our experiments involve two datasets (Table 1):

• **emrQA** (Pampari et al., 2018) is a large-scale clinical QA dataset, which was semi-automatically generated based on medical expert-made question templates and existing annotations on n2c2 challenge datasets (n2c2, 2006). The complete dataset consists of 5 subsets, and we use the "Relation" subset as it is the largest. Following the guidance in (Yue et al., 2020), we randomly sample 5% data as the actual training set since emrQA contains many redundant QA pairs.

• **MIMIC-III** (Johnson et al., 2016) is a large database covering patients who stayed in the Beth Israel Deaconess Medical Center. We randomly sample a set of discharge summaries (a type of clinical texts) from MIMIC-III, which only contain raw clinical texts and do not have any annotated QA pairs. We notice that there are a small portion of overlapped clinical texts between MIMIC-III and emrQA. When sampling MIMIC-III clinical texts, we ensure that all the sampled clinical texts do not appear in the emrQA.

We consider the emrQA dataset as *source* and the MIMIC-III dataset as *target*. Since there are no annotated QA pairs on the MIMIC-III dataset, we ask three clinical experts to create around 1,200 QA pairs as the test set, *and will seek to release them under MIMIC-III license.*

Specifically, sampled MIMIC-III clinical texts are given to the clinical experts, based on which, they can ask any questions as long as an answer can be extracted from the context.

To save annotation efforts, machine-generated QA pairs by 9 QG models (i.e., all base QG models and their variants; see Section 4.2) are provided as references. However, they are highly encouraged to create new questions based on the given clinical text (which are marked as *"human-generated"*). But if they do find the machine-generated questions make sense, sound natural and match the associated answer evidence, they can keep them (which are marked as *"human-verified"*). After obtaining the annotated questions, we ask another clinical expert to do a final pass of the questions in order to further guarantee the quality of the test set. The final test set consists of 1287 questions (of which 975 are *"human-verified"* and 312 are *"human-generated"*). In the experiment, we show the QA performance on both human-verified questions and human-generated ones (see Table 3).

Table 1: Statistics of the datasets. We synthesize a machine-generated dev set and ask human experts to annotate a test set for MIMIC-III.

(Question / Context)	emrQA	MIMIC-III
# Train	781,857 / 337	- / 337
# Dev	86,663 / 41	8,824 / 40
# Test	98,994 / 42	1,287 / 36
# Total	967,514 / 420	- / 413
for purpose of	QG & QA (source)	QA (target)

The dev set of MIMIC-III is constructed by sampling generated questions from 9 QG models [3] and is used to tune the hyper-parameters.

4.2 Base QG and QA models

We instantiate our `CliniQG4QA` framework using three base QG models:

• **NQG** (Du et al., 2017) is the first seq2seq model with a global attention mechanism (Luong et al., 2015) for question generation.

• **NQG++** (Zhou et al., 2017) is one of the most commonly adopted QG baselines with a feature-enriched encoder (e.g., lexical features) and a copy mechanism (Gulcehre et al., 2016).

• **BERT-SQG** (Chan and Fan, 2019) uses a pretrained BERT model (we use ClinicalBERT (Alsentzer et al., 2019) to accommodate clinical setting) as the encoder and formulates the decoding as a "MASK" token prediction problem.

To investigate the effectiveness of our QPP module, we consider the following variants of base QG models: (1) Base Model: Inference with greedy search; (2) Base Model + Beam Search: Inference with Beam Search with the beam size at K and keep Top K beams (a larger beam size hurts performance in our preliminary experiments, so we set K to 3); (3) Base Model + QPP: Inference with greedy search for both QPP module and Base model. Since our QPP module predicts a set of question phrases, Base Model + QPP can generate diverse questions.

[3] Instead of uniformly sampling from 9 QG models, we followed the sampling ratio of 1:3:6 (Base model, Base+BeamSearch, Base+QPP) for each QG method, which made the dev set cover as many diverse questions as possible.

Table 2: Automatic evaluation of the generated questions on emrQA dataset. For each base model, the best performing variant is in **bold**. RG: ROUGE-L, MR: METEOR, Dist: Distinct, Ent: Entropy.

Models	Relevance				Diversity			
	BLEU3	BLEU4	MR	RG	Dist3	Dist4	Ent3	Ent4
NQG (Du et al., 2017)	91.45	90.11	60.70	94.62	0.233	0.282	4.473	4.738
+ BeamSearch	94.33	93.42	62.08	95.56	0.569	0.775	5.406	5.812
+ QPP (Ours)	**96.82**	**96.33**	**64.38**	**97.49**	**3.177**	**5.289**	**7.100**	**7.777**
NQG++ (Zhou et al., 2017)	97.11	96.65	71.57	97.86	0.229	0.275	4.419	4.648
+ BeamSearch	98.35	98.07	72.98	98.55	0.618	0.848	5.497	5.953
+ QPP (Ours)	**99.15**	**99.03**	**74.01**	**99.11**	**3.183**	**5.293**	**7.111**	**7.798**
BERT-SQG (Chan and Fan, 2019)	89.07	87.99	65.25	94.91	0.228	0.276	4.594	4.849
+ BeamSearch	95.45	94.84	66.39	96.22	0.510	0.713	5.522	6.015
+ QPP (Ours)	**96.54**	**96.19**	**67.51**	**97.42**	**3.344**	**5.332**	**7.173**	**7.816**

For QA, we instantiate `CliniQG4QA` with two base models, DocReader (Chen et al., 2017) and ClinicalBERT (Alsentzer et al., 2019). Note that more complex QG/QA models and training strategies can also be used in our framework. As this work focuses on exploring how *diverse* questions help QA on target contexts, we adopt fundamental QG/QA models and training strategies, and leave more advanced ones that are complementary to our framework as future work.

4.3 Evaluation Metrics

For QG evaluation, we focus on evaluating both relevance and diversity. Following previous works (Du et al., 2017; Zhang et al., 2018), we use BLEU (Papineni et al., 2002), ROUGE-L (Lin, 2004) as well as METEOR (Lavie and Denkowski, 2009) for relevance evaluation. Since the Beam Search and our QPP module enable QG models to generate multiple questions given a evidence, we report the top-1 relevance among the generated questions following (Cho et al., 2019). For diversity, we report Distinct (Li et al., 2016) as well as Entropy (Zhang et al., 2018) scores. We calculate BLEU and the diversity measures based on 3- and 4-grams.

For QA evaluation, we report exact match (EM) (the percentage of predictions that match the ground truth answers exactly) and F1 (the average overlap between the predictions and ground truth answers) as in (Rajpurkar et al., 2016).

Implementation details. We provide implementation details for reproducing in Appendix B.

5 Experimental Results

5.1 Can QPP Encourage Diverse Questions?

Automatic Evaluation. We first evaluate QG models on the emrQA dataset. As can be seen from

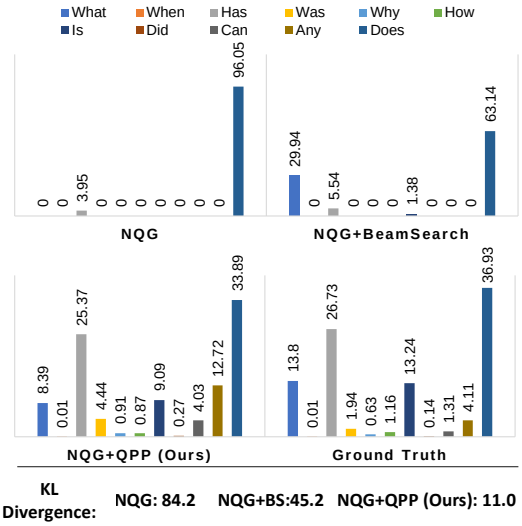

Figure 2: Distributions over types of questions generated by NQG models and the ground truth.

Table 2, the three selected base models (NQG, NQG++ and BERT-SQG) all achieve very promising relevance scores; however, they do not perform well with diversity scores. The diversity of generated questions is boosted to some extent when the Beam Search is used since it can offer flexibility for QG models to explore more candidates when decoding. In comparison, the QPP module in our framework leads to the best results under both relevance and diversity evaluation. Particularly, it obtains 5% absolute improvement in terms of Dist4 for each base model.

A Closer Look at Generated Question Types. To further demonstrate QPP module can help generate diverse questions, we show the distribution over each type of questions generated by NQG-based models and the ground truth in Figure 2.

We observe that the Kullback–Leibler (KL) divergence between the distributions of generated questions and the ground truth is the smallest after enabling our QPP module. Even some of the least

Table 3: The QA performance on MIMIC-III test set. emrQA is also included as a baseline dataset to help illustrate the generated diverse questions on MIMIC-III are useful to improve the QA model performance on new contexts.

QA Datasets	DocReader (Chen et al., 2017)						ClinicalBERT (Alsentzer et al., 2019)					
	Human Verified		Human Generated		Overall Test		Human Verified		Human Generated		Overall Test	
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
emrQA (Pampari 2018)	61.44	78.82	69.87	83.66	63.48	79.99	61.23	78.56	69.23	82.83	63.17	79.59
NQG (Du 2017)	64.71	79.36	66.99	79.67	65.26	79.43	59.49	76.68	67.3	82.59	61.38	78.11
+ BeamSearch	67.07	81.21	71.15	83.07	68.07	81.66	63.17	79.17	68.91	84.26	64.56	80.4
+ QPP (Ours)	**68.82**	**82.89**	**74.68**	**85.18**	**70.09**	**83.44**	**63.79**	**79.56**	**69.23**	**84.33**	**65.11**	**80.72**
NQG++ (Zhou 2017)	65.94	78.71	66.34	81.34	66.04	79.35	59.59	75.85	65.06	80.11	60.92	76.88
+ BeamSearch	68.10	80.09	72.11	84.56	69.07	81.17	64.61	80.30	68.26	83.70	65.50	81.12
+ QPP (Ours)	**70.05**	**83.47**	**74.36**	**85.92**	**71.10**	**84.06**	**65.33**	**80.64**	**70.83**	**85.76**	**66.67**	**81.88**
BERT-SQG (Chan 2019)	66.05	79.64	70.19	81.47	67.05	80.08	59.59	78.04	65.06	82.20	60.92	79.05
+ BeamSearch	68.71	81.98	73.71	84.44	69.93	82.58	61.94	79.02	67.31	82.54	63.25	79.88
+ QPP (Ours)	**70.77**	**83.60**	**74.36**	**85.53**	**71.64**	**84.07**	**64.21**	**80.53**	**69.23**	**85.38**	**65.43**	**81.71**

frequent types of questions (e.g., “How", “Why") can be generated. Similar results are also observed in NQG++ and BERT-SQG based models (in Appendix Table A5), which demonstrates QPP module can help generate diverse questions.

5.2 Can Generated Questions Help QA on New Contexts?

Automatic Evaluation. Table 3 summarizes the performance of two widely used QA models, DocReader (Chen et al., 2017) and ClinicalBERT (Alsentzer et al., 2019), on the MIMIC-III testing set. The QA models are trained based on different corpora, including the emrQA dataset as well as QA pairs generated by different models. For a fair comparison, we keep the total number of generated QA pairs roughly the same as emrQA. As can be seen from the table, the QA models based on the corpora that are generated using the three base QG models can only achieve roughly the same or even worse performance compared with the QA models trained on the emrQA dataset. Though the Beam Search strategy could boost the diversity to some extent and lead the improvement of QA models, our proposed QPP module can improve QA model training. For example, training DocReader using questions generated by NQG++ with our QPP module outperforms that using the emrQA dataset by around 8% under EM and 4% under F1 on the overall test set. Moreover, the results on human-generated portion are consistently better than that on human-verified. It’s attributed to the fact that human-created questions are more readable and sensible while human-verified questions are a bit of less natural though correctness is ensured.

All these results indicate that generating a diverse QA corpus is useful for the downstream QA

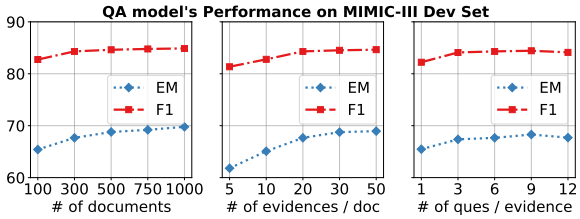

Figure 3: Influence of the number of documents, number of evidences per document, number of QA pairs per evidence on QA performance.

on the new context, and our simple QPP module can help existing QG models achieve such a goal.

Why QG Boosts QA on New Contexts? To further explore why QG can boost QA, we consider three major factors when generating a QA corpus: the number of documents, the number of answer evidences per document, and the number of generated questions per answer evidence. When we test one factor, we fix the other two. For example, we fix the number of answer evidences and questions at 20 and 6 when we test the influence of the number of documents. We use NQG++ and DocReader as our base QG and QA models to instantiate our `CliniQG4QA` framework and report the performance on the Dev set.

As can seen from Figure 3, the performance steadily increases when we use more documents and more answer evidences during QA corpus generation. This can demonstrate the first hypothesis: The generated corpus enables a QA model to see more new contexts during training, which can help the QA model get a better understanding of similar contexts during testing. The more contexts it sees, the more benefits it could obtain. We can also see that with the increase of the number of generated questions per evidence, the performance generally rises up. This indicates that multiple diverse questions are essential for boosting QA performance.

QA Example from MIMIC-III

Context: ... he was guaiac negative on admission. hematocrit remained stable overnight. **5. abd pain: suspect secondary to chronic pancreatitis.** amylase unchanged from previous levels. ...

Question: Why did the patient get abd pain?
Answer by QA model trained on
-*emrQA:* 5. abd pain
-*NQG Generated:* 5. abd pain:
-*NQG+BeamSearch:* 5. abd pain:
-NQG+QPP: 5. abd pain: suspect secondary to chronic pancreatitis**.**

QG Example from MIMIC-III

Context: ... the patient was taking at home prior to admission were not restarted. **25. acetaminophen 325-650 mg po/ng q6h:prn pain** 26. dabigatran etexilate 150 mg po bid...

Questions generated by
-*NQG:* **Does** the patient have any pain?
-*NQG+BeamSearch:* **Does** the patient have any pain history? **Does** the patient have pain? **Does** the patient have any pain?
-NQG+QPP: **Why** did the patient have acetaminophen? **What** treatment has the patient had for his pain? **How** was pain treated? **Does** the patient have any pain? ...

Figure 4: QA and QG examples. The red parts in contexts are ground-truth answer evidences.

In summary, we think seeing many new contexts and diverse questions are the two main reasons why QA models are boosted.

Diverse Questions Really Matter for QA: Two Real Cases. In Figure 4, we present a QA example and a QG example from MIMIC-III.

In the QA example, this "why" question can be correctly answered by the QA model (DocReader) trained on the "NQG+QPP" generated corpus while the QA models trained on other generated corpora fail. This is because the NQG model and "NQG+BeamSearch" cannot generate any "why" questions as shown in Figure 2. Thus QA models trained on such corpora cannot answer questions of less frequent types. Though the emrQA dataset contains diverse questions (including "why" questions), its contexts might be different from MIMIC-III in terms of topic, note structures, writing styles, etc. So the model trained on emrQA struggles to answer some questions.

In the QG example, the base model NQG can only generate one question. Though utilizing the Beam Search enables the model to explore multiple candidates, the generated questions are quite similar and are less likely to help improve QA. Enabling our QPP module helps generate diverse questions including "Why", "What", "How", etc.

5.3 Ablation Study

Impact of Question Phrase Length. We show the influence of Question Phrase Length on the QG in Table 4. We set question phrase length at 2 since it helps the model achieve the best performance overall, and allows the model to generate both relevant and diverse questions as well as have a smaller search space.

Table 4: Impact of question phrase length (l) on QG model performance (on emrQA dev set).

Models	Relevance		Diversity	
	BLEU4	**MR**	**Dist4**	**Ent4**
NQG++	97.36	72.52	0.313	4.779
+QPP (l=1)	99.08	72.67	2.228	7.092
+QPP (l=2)	**99.10**	**74.51**	**5.554**	7.801
+QPP (l=3)	**99.10**	73.58	5.157	**8.026**

Table 5: Choosing seq2seq-based QPP over alternative multi-label classification methods. BR: Binary Relevance; CC: Classifier Chain; HL: Hamming Loss.

Models	**HL**	**P**	**R**	**F1**
BR	0.0524	**99.22**	90.89	94.87
CC	0.0524	**99.22**	90.89	94.87
QPP	**0.0346**	97.28	**96.20**	**96.74**

Alternative Approaches for QPP. There are many model options for the QPP task, e.g., those for multi-label classification. To justify our choice of a seq2seq model, we compare it with two commonly-adopted multi-label classification (MLC) methods based on binary relevance (BR) and classifier chain (CC) (Boutell et al., 2004; Read et al., 2011). BR develops multiple binary classifiers independently while CC builds a chain of classifiers and predicts labels sequentially. We use multi-layer perceptron as the specific model architecture for both BR and CC. For each answer evidence, the input is the representation from the same LSTM encoder as our QPP module.

From Table 5, we can see: (1) The sequential decoder in our current QPP module performs better overall and especially in terms of Recall, which is particularly important since we aim for generating diverse question types; (2) A simple seq2seq model achieves great performance across all metrics, which renders developing more complex models for this task less necessary.

6 Conclusion

This paper proposes a simple yet effective framework for improving clinical QA on new contexts. It leverages a seq2seq-based question phrase prediction (QPP) module to enable QG models to generate diverse questions. Our comprehensive experiments and analyses allow for a better understanding of why diverse question generation can help QA on new clinical documents.

Acknowledgements

This research was sponsored in part by the Patient-Centered Outcomes Research Institute Funding ME-2017C1-6413, the Army Research Office under cooperative agreements W911NF-17-1-0412, NSF Grant IIS1815674, NSF CAREER #1942980, and Ohio Supercomputer Center (Center, 1987). The views and conclusions contained herein are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S.Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notice herein.

References

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. 2019. Synthetic qa corpora generation with roundtrip consistency. In *ACL'19*, pages 6168–6173.

Emily Alsentzer, John R Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical bert embeddings. *NAACL Clinical NLP Workshop 2019*.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175.

Matthew R. Boutell, Jiebo Luo, Xipeng Shen, and Christopher M. Brown. 2004. Learning multi-label scene classification. *Pattern Recognition*, page 1757–1771.

Brian L Cairns, Rodney D Nielsen, James J Masanz, James H Martin, Martha S Palmer, Wayne H Ward, and Guergana K Savova. 2011. The mipacq clinical question answering system. In *AMIA annual symposium proceedings*, volume 2011, page 171.

Yu Cao, Meng Fang, Baosheng Yu, and Joey Tianyi Zhou. 2020. Unsupervised domain adaptation on reading comprehension. In *AAAI'20*.

Ohio Supercomputer Center. 1987. Ohio supercomputer center.

Ying-Hong Chan and Yao-Chung Fan. 2019. A recurrent bert-based model for question generation. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 154–162.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *ACL'17*, pages 1870–1879.

Jaemin Cho, Minjoon Seo, and Hannaneh Hajishirzi. 2019. Mixture content selection for diverse sequence generation. In *EMNLP-IJCNLP'19*, pages 3112–3122.

Yu-An Chung, Hung-Yi Lee, and James Glass. 2018. Supervised and unsupervised transfer learning for question answering. In *NAACL-HLT'18*, pages 1585–1594.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT'19*, pages 4171–4186.

Xinya Du and Claire Cardie. 2018. Harvesting paragraph-level question-answer pairs from wikipedia. In *ACL'18*, pages 1907–1917.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. In *ACL'17*, pages 1342–1352.

Jungwei Fan. 2019. Annotating and characterizing clinical sentences with explicit why-qa cues. In *NAACL Clinical NLP Workshop*, pages 101–106.

David Golub, Po-Sen Huang, Xiaodong He, and Li Deng. 2017. Two-stage synthesis networks for transfer learning in machine comprehension. In *EMNLP'17*, pages 835–844.

Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the unknown words. In *ACL'16*, pages 140–149.

Timothy J Hazen, Shehzaad Dhuliawala, and Daniel Boies. 2019. Towards domain adaptation from limited data for question answering using deep neural networks. *arXiv preprint arXiv:1911.02655*.

Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3:160035.

Junmo Kang, Haritz Puerto San Roman, et al. 2019. Let me know what to ask: Interrogative-word-aware question generation. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 163–171.

Alon Lavie and Michael J Denkowski. 2009. The meteor metric for automatic evaluation of machine translation. *Machine translation*, 23(2-3):105–115.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *NAACL-HLT'16*, pages 110–119.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Bang Liu, Haojie Wei, Di Niu, Haolan Chen, and Yancheng He. 2020. Asking questions the human way: Scalable question-answer generation from text corpus. In *WWW'20*, pages 2032–2043.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *EMNLP'15*, pages 1412–1421.

n2c2. 2006. n2c2 nlp research data sets.

Preksha Nema, Akash Kumar Mohankumar, Mitesh M Khapra, Balaji Vasan Srinivasan, and Balaraman Ravindran. 2019. Let's ask again: Refine network for automatic question generation. In *EMNLP-IJCNLP'19*, pages 3305–3314.

Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. 2019. Scispacy: Fast and robust models for biomedical natural language processing. In *ACL BioNLP Workshop 2019*.

Anusri Pampari, Preethi Raghavan, Jennifer Liang, and Jian Peng. 2018. emrqa: A large corpus for question answering on electronic medical records. In *EMNLP'18*, pages 2357–2368.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL'02*, pages 311–318. Association for Computational Linguistics.

Jon Patrick and Min Li. 2012. An ontology for clinical questions about the contents of patient notes. *Journal of Biomedical Informatics*, 45(2):292–306.

Tomasz Kajdanowicz Piotr Szymański. 2017. A scikit-based python environment for performing multi-label classification. *arXiv preprint arXiv:1702.01460*.

Preethi Raghavan, Siddharth Patwardhan, Jennifer J Liang, and Murthy V Devarakonda. 2018. Annotating electronic medical records for question answering. *arXiv preprint arXiv:1805.06816*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *EMNLP'16*, pages 2383–2392.

Lance A Ramshaw and Mitchell P Marcus. 1999. Text chunking using transformation-based learning. In *Natural language processing using very large corpora*, pages 157–176. Springer.

Bhanu Pratap Singh Rawat, Wei-Hung Weng, Preethi Raghavan, and Peter Szolovits. 2020. Entity-enriched neural models for clinical question answering. *arXiv preprint arXiv:2005.06587*.

Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. 2011. Classifier chains for multi-label classification. *Machine learning*, 85(3):333.

Kirk Roberts and Braja Gopal Patra. 2017. A semantic parsing method for mapping clinical questions to logical forms. In *AMIA Annual Symposium Proceedings*, volume 2017, page 1478.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hananneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *ICLR'17*.

Sheng Shen, Yaliang Li, Nan Du, Xian Wu, Yusheng Xie, Shen Ge, Tao Yang, Kai Wang, Xingzheng Liang, and Wei Fan. 2020. On the generation of medical question-answer pairs. In *AAAI*, pages 8822–8829.

Sarvesh Soni, Meghana Gudala, Daisy Zhe Wang, and Kirk Roberts. 2019. Using fhir to construct a corpus of clinical questions annotated with logical forms and answers. In *AMIA'19*, volume 2019, page 1207.

Sarvesh Soni and Kirk Roberts. 2019. A paraphrase generation system for ehr question answering. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 20–29.

Sarvesh Soni and Kirk Roberts. 2020. Paraphrasing to improve the performance of electronic health records question answering. *AMIA Summits on Translational Science Proceedings*, 2020:626.

Xingwu Sun, Jing Liu, Yajuan Lyu, Wei He, Yanjun Ma, and Shi Wang. 2018. Answer-focused and position-aware neural question generation. In *EMNLP'18*, pages 3930–3939.

Luu Anh Tuan, Darsh J Shah, and Regina Barzilay. 2020. Capturing greater context for question generation. In *AAAI'20*.

Huazheng Wang, Zhe Gan, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, and Hongning Wang. 2019. Adversarial domain adaptation for machine reading comprehension. In *EMNLP-IJCNLP'19*, pages 2510–2520.

Andrew Wen, Mohamed Y Elwazir, Sungrim Moon, and Jungwei Fan. 2020. Adapting and evaluating a deep learning language model for clinical why-question answering. *JAMIA Open*, 3(1):16–20.

Georg Wiese, Dirk Weissenborn, and Mariana Neves. 2017. Neural domain adaptation for biomedical question answering. In *CoNLL'17*, pages 281–289.

Zhilin Yang, Junjie Hu, Ruslan Salakhutdinov, and William Cohen. 2017. Semi-supervised qa with generative domain-adaptive nets. In *ACL'17*, pages 1040–1050.

Xiang Yue, Bernal Jimenez Gutierrez, and Huan Sun. 2020. Clinical reading comprehension: A thorough analysis of the emrqa dataset. In *ACL'20*.

Shiyue Zhang and Mohit Bansal. 2019. Addressing semantic drift in question generation for semi-supervised question answering. In *EMNLP-IJCNLP'19*, pages 2495–2509.

Yizhe Zhang, Michel Galley, Jianfeng Gao, Zhe Gan, Xiujun Li, Chris Brockett, and Bill Dolan. 2018. Generating informative and diverse conversational responses via adversarial information maximization. In *NeurIPS'18*, pages 1810–1820.

Yao Zhao, Xiaochuan Ni, Yuanyuan Ding, and Qifa Ke. 2018. Paragraph-level neural question generation with maxout pointer and gated self-attention networks. In *EMNLP'18*, pages 3901–3910.

Qingyu Zhou, Nan Yang, Furu Wei, Chuanqi Tan, Hangbo Bao, and Ming Zhou. 2017. Neural question generation from text: A preliminary study. In *National CCF Conference on Natural Language Processing and Chinese Computing*, pages 662–671. Springer.

A Post-Processing of Extracted Answer Evidences

This section is dedicated to the discussion of heuristic rules developed to improve the quality of the extracted answer evidence.

We observe that when we directly apply the ClinicalBERT (Alsentzer et al., 2019) system described in Section 3.2 (in the main content) on clinical texts, the extracted answer evidences sometimes are broken sentences due to the noisy nature and uninformative language (e.g., acronyms) of clinical texts. To make sure the extracted evidences are meaningful, we designed a *"merge-and-drop"* heuristic rule to further improve the extractor's accuracy. Specifically, for each extracted evidence candidate, we first examine the *length* (number of tokens) of the extracted evidence. If the length is larger than the threshold η, we keep this evidence; otherwise, we compute the *distance*, i.e., the number of tokens between the current candidate span and the closest span. If the *distance* is smaller than the threshold γ, we merge these two "close-sitting" spans; otherwise, we drop this over-short evidence span. In our experiments, we set η and γ to be 3 and 3, respectively, since they help the QA system achieve the best performance on the dev set

The intuitions behind this heuristic rule are listed as follows: 1) we should discard as few useful answer evidence spans as possible, so we first resort to merging before simply dropping a span; 2) Commonly, if two spans (and at least one is a short span) are sitting close to each other, they should have been recognized as a single answer evidence. For example, the `BIO` label of a snippet "chief complaint: altered mental status major" is predicted as "B O B I I O", whereas, a clinical expert will label it as "B I I I I O". However, if they sit far away, merging would introduce noisy information; 3) long spans are always more informative and contain less misleading information compared with short spans (e.g., "the patient had left leg pain" v.s. "pain").

B Implementation Details

We provide very detailed implementation details to foster reproducibility.

	QPP acc	QG acc
NQG	/	74.32
NQG++	/	74.54
BERT-SQG	/	79.78
QPP-NQG	99.2	85.15
QPP-NQG++	99.17	85.27
QPP-BERT-SQG	99.19	88.15

Table A1: QPP and QG performance on dev set in terms of per-token accuracy. All numbers are percentages.

`CliniQG4QA`	QG	QA	overall
NQG&DocReader	6	17	24.5
NQG++&DocReader	7.5	20	29
BERT-SQG&DocReader	13	18	32.5
NQG&CliniBERT	6	3.5	11
NQG++&CliniBERT	7.5	4	13
BERT-SQG&CliniBERT	13.5	3.5	18.5

Table A2: The running time (hour) of QPP-augmented QG, QA and overall `CliniQG4QA` based on our selected QG & QA combinations

B.1 Prepossessing

B.1.1 Dataset Prepossessing

emrQA[4] We prune the emrQA set by removing QA pairs whose question is an indicator (e.g., "meds") so that all remaining QA pairs contain valid questions. We also leverage SciSpacy[5], a package containing spaCy models for processing clinical text, to do tokenization in order for the trained QG models to have a better understanding over clinical notes.

MIMIC-III[6] Similarly, we also leverage SciSpacy to do tokenization.

B.1.2 Question Phrases Identification

In order to utilize our Question Phrase Prediction (QPP) module and make our QPP module generic enough without loss of generality, we identify valid n-gram Question Phrases in an automatic way.

To prepare an exhaustive list of valid n-gram Question Phrases, we first collect all of the first n words appearing in Ground Truth Questions in emrQA, forming three (i.e., n=1, 2, 3) raw Question Phrases set.

We observe that all uni-grams are valid question phrases (e.g., "How", "When", "What"), so we don't do any pruning and keep the uni-gram question phrases set as it is.

[4] https://github.com/panushri25/emrQA
[5] https://allenai.github.io/scispacy/
[6] https://mimic.physionet.org/gettingstarted/access/

Parameter	Search Trials	Best
Dropout	[0.3,0.45,0.6,0.75,0.9]	0.75
LSTM Layers	[1,2,3,4]	3
QP Length	[1,2,3]	2

Table A3: Hyperparameter searches for Question Phrase Prediction (QPP) Module

Parameter	Search Trials	Best
length (η)	[1,3,5,10]	3
distance (γ)	[1,3,5,10]	3

Table A4: Hyperparameter searches for *"merge-and-drop"* method

As for n-gram (n $\geq$ 2) Question Phrases set, we conduct fine-grained filtering. We only consider n-grams with occurrence frequency greater than the threshold ζ as valid n-gram Question Phrases. In our experiment, we set ζ as 0.02%. Less frequent n-gram words (i.e., frequency $<$ 0.02%) will degrade to unigram Question Phrases in accordance with corresponding question types (e.g., "Has lasix" $\rightarrow$ "Has"*) so as to maintain lossless. In the end, n-gram (n $\geq$ 2) Question Phrases sets, without any information loss, are consisting of both n-gram Question Phrases and degraded unigram Question Phrases.

B.2 Models Implementation

Base QA & QG Models We re-implement the three base QG models using Pytorch and have ensured that they achieve comparable performance as originally reported. The best QG model is selected using the per-token accuracy of both the QPP module (if applicable) and QG on dev set, and dev results are listed in Table A1.

For QA models, we used their open-sourced implementation.[7] The best QA model is selected using EM and F1 on dev set, and dev results are also included in Table 3 (in the main content). Hyperparameters of QG models are set to be the same as in the original paper and hyperparameters of QA models are set according to the guidance of (Yue et al., 2020).

Question Phrase Prediction (QPP) Module. Word embeddings are initialized by Glove 300d vectors[8]. We adopt a feature-rich Encoder[9] to effectively encode Clinical lexical information. We set the LSTM hidden unit size to 600 and set the number of layers of LSTMs to 3 in both encoder and decoder. Optimization is performed using stochastic gradient descent (SGD) for 20 epochs, with an initial learning rate of 1.0. After each epoch, we evaluate the per-label accuracy on the dev set. If the accuracy does not improve, we halve the learning rate. The mini-batch size is set at 128. Dropout with probability 0.75 is applied between vertical LSTM layers. The gradient is clipped when its norm exceeds 5. Besides, we set the length of a question phrase l to 2, which gives the best performance on validation. The total number of parameters is around 17M under our best-performing setting.

Answer Evidence Extractor. We fine-tune a ClinicalBERT model in Named Entity Recognition (NER) fashion using `BIO` tagging scheme[10]. When conducting fine-tuning on our QG train set, we set max length, batch size, number of epochs, and random seed to be 510, 16, 20 and 6, respectively. We adopt the official NER evaluation script[11] to do the evaluation on QG dev set, and obtained 80.17 F1 score. We then deployed this system to extract raw answer evidence spans. After raw extraction, we utilized our own heuristic rules (i.e., *"merge-and-drop"*) to further polish the raw spans as described in Appendix A.

Multi-Label Classification (MLC) Comparison. We implement Binary Relevance (BR) and Classifier Chain (CC) by means of Scikit-Multilearn (Piotr Szymański, 2017), an open-source library for the MLC task.

Computational Resources. All experiments are conducted using one single GeForce GTX 2080 Ti 12 GB GPU (with significant CPU resources). We train the QPP module and QG model together though they can be trained separately. The overall running time of our `CliniQG4QA` system depends on the particular QG and QA models adopted. For our selected QG and QA models, the approximated overall running time of `CliniQG4QA` is listed in Table A2. However, training a QPP module separately is fast, which only takes less than 1 hour with the current setting. Meanwhile, the running time of our Answer Evidence Extractor roughly takes 1.5 hours on average.

[7] DocReader: https://github.com/facebookresearch/DrQA. ClinicalBERT: https://github.com/EmilyAlsentzer/clinicalBERT.

[8] http://nlp.stanford.edu/data/glove.840B.300d.zip

[9] Lexical features are extracted by (Neumann et al., 2019)

[10] https://github.com/huggingface/transformers/tree/master/examples/token-classification

[11] http://deeplearning.net/tutorial/code/conlleval.pl

Table A5: Distributions of the generated questions of different models and the ground truth in the emrQA dataset. QPP: Question Phrase Prediction; KL: Kullback–Leibler divergence. All numbers are percentages.

Models	**What**	**When**	**Has**	**Was**	**Why**	**How**	**Is**	**Did**	**Can**	**Any**	**Does**	**KL (Gen||GT)**
NQG	0.00	0.00	3.95	0.00	0.00	0.00	0.00	0.00	0.00	0.00	96.05	84.2
+BeamSearch	29.94	0.00	5.54	0.00	0.00	0.00	1.38	0.00	0.00	0.00	63.14	45.2
+QPP (Ours)	8.39	0.01	25.37	4.44	0.91	0.87	9.09	0.27	4.03	12.72	33.89	**11.0**
NQG++	0.09	0.00	3.53	0.00	0.00	0.10	0.00	0.00	0.00	0.00	96.28	84.3
+BeamSearch	44.04	0.00	0.09	0.00	0.00	0.22	2.01	0.00	0.00	0.00	53.64	66.4
+QPP (Ours)	8.09	0.01	25.54	4.42	0.75	0.81	9.16	0.23	4.05	12.80	34.13	**11.2**
BERT-SQG	0.72	0.00	6.32	0.00	0.00	0.00	0.00	0.00	0.00	0.00	92.96	74.5
+BeamSearch	29.31	0.00	0.03	0.00	0.00	0.00	25.05	0.00	0.00	0.00	45.62	47.4
+QPP (Ours)	8.01	0.01	25.56	4.44	0.82	0.80	9.17	0.25	4.05	12.74	34.16	**11.2**
GT	13.80	0.01	26.73	1.94	0.63	1.16	13.24	0.14	1.31	4.11	36.93	-

B.3 Hyperparameter Search

In order to have a best-performing Question Phrase Prediction (QPP) module, we manually tuned the hyperparameters listed in Table A3. The hyperparameters are tuned on QG dev set using Relevance and Diversity Metrics listed in Section 4.3 (in the main content).
In order to have a best-performing post-processing method (i.e., *"merge-and-drop"*) in Answer Evidence Extraction module, we manually tuned the hyperparameters listed in Table A4. The hyperparameters are tuned on the QA dev set using Exact Match (EM) and F1.

C Distributions of Generated Questions of Different QG Models

The detailed distributions of the generated questions of different models and the ground truth in emrQA dataset are listed in table A5.