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

Xiang Yue[1,*], Xinliang Frederick Zhang[1,2,*], Ziyu Yao[1,3], Simon Lin[4], and Huan Sun[1]

[1]The Ohio State University
[2]University of Michigan
[3]George Mason University
[4]Abigail Wexner Research Institute at Nationwide Children's Hospital
{yue.149, zhang.9975, sun.397}@osu.edu
ziyuyao@gmu.edu
Simon.Lin@nationwidechildrens.org
[*]These two authors contributed equally

***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 model 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 the generation. Our comprehensive experiment results show that the QA corpus generated by our framework can 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]

***Index Terms*—Clinical Question Answering, Clinical Question Generation, Natural Language Processing, Domain Adaptation, Clinical Text**

I. 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 [1]–[5]. Neural QA models in recent years [5]–[7] 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 [8], when a clinical QA model that was trained on the emrQA dataset [3] is deployed to answer questions based on MIMIC-III clinical texts [9], 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 [10], to synthesize clinical QA pairs on target contexts to facilitate the QA model training (Figure 2). 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* (AEE) 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 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 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, which takes an answer evidence as input and sequentially predicts potential question phrases (e.g., "What treatment", "How often") that signify what types of

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

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

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.

Due to the lack of publicly-available clinical QA pairs for our proposed domain adaptation evaluation setting, we ask clinical experts to annotate a new test set on the sampled MIMIC-III [9] clinical texts. We conduct extensive experiments to evaluate `CliniQG4QA`, using emrQA [3] as the source contexts and our annotated MIMIC-III [9] 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 show that the proposed QPP module can substantially help generate much more diverse types of questions (e.g., "When" and "Why" questions). More importantly, we systematically demonstrate the strong capability of `CliniQG4QA` for improving QA performance on new contexts by evaluating it on our constructed MIMIC-III QA dataset. 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.

II. Preliminary and Related Work

Clinical Question Answering aims to extract a text span (a sentence or multiple sentences) as the answer from a patient clinical note given a question (Fig. 1 left) [8]. Though many neural models [5]–[7], [11], [12] 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 [8]. 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. Moreover, during the pandemic, clinical QA models can also be deployed to answer COVID-19 related questions [13], [14].

Question Generation seeks to automatically generate questions given a sentence or paragraph (Fig. 1 right). Existing QG models [10], [15]–[23] 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 [24]–[26] explores 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 [3]), rendering the difficulty of directly deploying their model on the clinical QA.

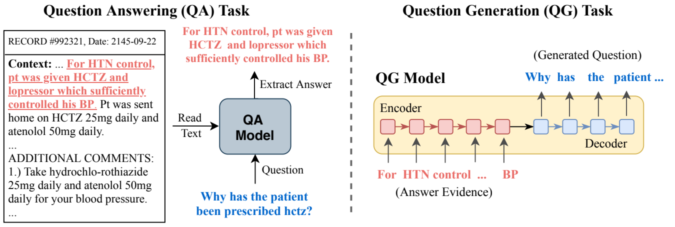
Question Answering (QA) Task
RECORD #992321, Date: 2145-09-22
Context: ... For HTN control, pt was given HCTZ and lopressor which sufficiently controlled his BP. Pt was sent home on HCTZ 25mg daily and atenolol 50mg daily.
...
ADDITIONAL COMMENTS:
1.) Take hydrochlo-rothiazide 25mg daily and atenolol 50mg daily for your blood pressure.
...
Read Text
QA Model
For HTN control, pt was given HCTZ and lopressor which sufficiently controlled his BP.
Extract Answer
Question
Why has the patient been prescribed hctz?
Question Generation (QG) Task
(Generated Question)
QG Model
Why has the patient ...
Encoder
Decoder
For HTN control ... BP
(Answer Evidence)

Fig. 1: Illustration of Clinical Question Answering (QA) and Question Generation (QG) task.

In the clinical and medical domain, [27] and [28], [29] 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.

Our aim is to improve clinical QA on new clinical texts (i.e., domain adaptation of clinical QA). We assume the availability of a large set of QA pairs and corresponding clinical documents (source contexts), and our goal is to better answer questions on new documents (target contexts) where only unlabeled documents are provided. We leverage a QG model to synthesize diverse QA pairs to save medical experts annotation efforts and improve QA performance without requiring extra annotations. Our setting is very practical in the real-world scenario, since it is infeasible to always annotate QA pairs on new clinical texts when deploying a QA system into a new environment.

III. Methods

A. *Overview of Our Framework*

We first give an overview of our `CliniQG4QA` framework (Fig 2). `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 new contexts.

B. *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 [3], [8], 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

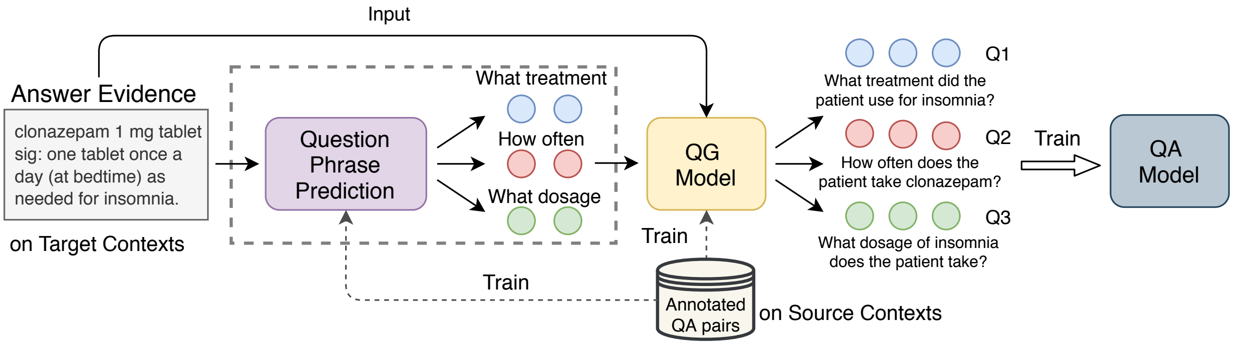
Input
Answer Evidence
clonazepam 1 mg tablet sig: one tablet once a day (at bedtime) as needed for insomnia.
on Target Contexts
Question Phrase Prediction
What treatment
How often
What dosage
QG Model
Train
Annotated QA pairs
on Source Contexts
Q1
What treatment did the patient use for insomnia?
Q2
How often does the patient take clonazepam?
Q3
What dosage of insomnia does the patient take?
Train
QA Model

Fig. 2: Illustration of our Question Phrase Prediction (QPP) module, which can be used together with QG models to diversify generations.

compared with short ones, which are very important for the clinical QA task.

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 [30], to label answer evidences.

Firstly, we adopt the ClinicalBERT model [31] to encode the document:

$$\mathbf{U} = \texttt{ClinicalBERT}\{p_1, ..., p_m\}. \tag{1}$$

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 [7], we use a linear layer on top of the hidden states output by ClinicalBERT followed by a softmax function to do the classification:

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

where a_j is the predicted BIO tag.

After prediction, we observe that 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 that 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 another closest candidate span If the *distance* is smaller than the threshold γ, we merge these two "close-sitting" spans; otherwise, we drop this overly-short evidence span. In our experiments, we set η and γ to be 3 and 3, respectively, since they help achieve the best performance on the dev set.

C. 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. We set $n = 2$ in our experiment as it achieves the best performance on the dev set. 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 $\mathbf{a}$. 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 [32] 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 in 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[3]. In the inference stage, QPP can dynamically decide the number of question phrases for each answer evidence by predicting a special [STOP] token. By decomposing QG into two steps (diversification followed by generation), the implemented QPP can increase the diversity in a more controllable way.

D. Training

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

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

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

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

[3] In our dataset, each answer evidence is tied with multiple questions, which allows the training for QPP.

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
  Train Answer Evidence Extractor based on the source data $\{(P_S, A_S)\}$ using Eq. 3
  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
  Train a QPP-enhanced QG model on the source data $\{(A_S, Y_S, Q_S)\}$ using Eq. 5
Training Stage
  Use AEE to extract potential answer evidences $\{A'_T\}$ on the target contexts $\{P_T\}$
  Use QPP to predict potential question phrases set $\{Y'_T\}$ on $\{A'_T\}$
  Use QPP-enhanced QG to generate diverse questions $\{Q'_T\}$ based on $\{(A'_T, Y'_T)\}$
  Train a QA model on synthetic target data $\{(P_T, A'_T, Q'_T)\}$ using Eq. 6
```

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

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

where θ denotes all the parameters of QPP.

Then we can train any QG model (e.g, NQG [10]) on source data by minimizing:

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

where μ denotes all parameters of the QG model.

During the *training* stage, given unlabeled target clinical documents, we first extract answer evidences, based on which QPP can be "plugged" into the QG model to generate diverse questions. Finally, a QA model (e.g., DocReader [6]) can be trained on the generated QA pairs of the target documents:

$$L_{QA} = -\sum_i \log P(\mathbf{a}|\mathbf{q},\mathbf{p};\delta) \tag{6}$$

where δ denotes all parameters of the QA model.

IV. Generalizability Test Set Construction

Unlike open domain, there are very few publicly available QA datasets in the clinical domain. EmrQA dataset [3], which was generated based on medical expert-made question templates and existing annotations on n2c2 challenge datasets [33], is a commonly adopted dataset for clinical reading comprehension.

However, all the QA pairs in emrQA are based on n2c2 clinical texts and thus not suitable for our generalization setting. [8] studied a similar problem and annotated a test set on MIMIC-III clinical texts [9]. However, their test set is too small (only 50 QA pairs) and not publicly available. Given the lack of a reasonably large clinical QA test set for studying generalization, with the help of three clinical experts, we create 1287 QA pairs on a sampled set of MIMIC-III [9] clinical notes, *which have been reviewed and approved by PhysioNet*[4] *and is downloadable by following the instructions*[5].

TABLE I: 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)

Annotation Process. We sample 36 MIMIC-III clinical notes as contexts. When sampling MIMIC-III notes, we ensure that all the sampled clinical texts do not appear in emrQA, acknowledging that there is a small overlap between the two datasets. For each context, clinical experts can ask any questions as long as an answer can be extracted from the context. To save annotation effort, QA pairs generated by QG models (i.e., all base QG models and their diversity-enhanced variants; see Section V-A) are provided as references, and duplicates are removed. Meanwhile, clinical experts are *highly encouraged* to create new questions based on the given clinical text (which are marked as *"human-generated"/"HG"*). But if they do find the machine-generated questions sound natural and match the provided answer, they can keep them (which are marked as *"human-verified"/"HV"*). After obtaining the annotated questions, we ask another clinical expert to do a final pass of the questions in order to further ensure 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"*).

We understand that there might be potential bias when evaluating QA models on the HV set (i.e, a QG model which is used to generate training questions for a QA model also contributes questions to the HV set as well). However, such bias might exist in human annotated data as well (e.g., the same set of humans create both training and testing dataset). Note that the contexts used to generate questions in HV/HG are separated from those to generate training questions. Besides, due to the relatively limited language patterns in clinical domain, we find most questions in HV set sound like what humans would ask. As such, we still deem it as a valuable asset and potential future research could leverage our HV set as their dev set to tune hyper-parameters.

To help tune the model, we also construct dev set of MIMIC-

[4] https://physionet.org/. PhysioNet is a resource center with missions to conduct and catalyze for biomedical research, which offers free access to large collections of physiological and clinical data, such as MIMIC-III [9].

[5] https://physionet.org/content/mimic-iii-question-answer/1.0.0/.

TABLE II: 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** [6]						**ClinicalBERT** [31]					
	Human Generated		**Human Verified**		**Overall Test**		**Human Generated**		**Human Verified**		**Overall Test**	
	EM	**F1**	**EM**	**F1**	**EM**	**F1**	**EM**	**F1**	**EM**	**F1**	**EM**	**F1**
emrQA [3]	69.87	83.66	61.44	78.82	63.48	79.99	69.23	82.83	61.23	78.56	63.17	79.59
NQG [10]	66.99	79.67	64.71	79.36	65.26	79.43	67.30	82.59	59.49	76.68	61.38	78.11
+ BeamSearch	71.15	83.07	67.07	81.21	68.07	81.66	68.91	84.26	63.17	79.17	64.56	80.40
+ Top-k Sampling	71.58	83.48	66.77	80.45	67.94	81.19	67.74	81.96	60.82	78.16	62.50	79.08
+ Nucleus Sampling	70.62	83.68	67.16	80.37	68.00	81.17	68.70	83.21	62.36	77.89	63.90	79.18
+ QPP (Ours)	**74.36**	**85.18**	**68.82**	**82.89**	**70.09**	**83.44**	**69.23**	**84.33**	**63.79**	**79.56**	**65.11**	**80.72**
NQG++ [15]	66.34	81.34	65.94	78.71	66.04	79.35	65.06	80.11	59.59	75.85	60.92	76.88
+ BeamSearch	72.11	84.56	68.10	80.09	69.07	81.17	68.26	83.70	64.61	80.30	65.50	81.12
+ Top-k Sampling	73.29	85.56	69.11	82.38	69.41	83.35	70.19	85.61	62.84	79.77	64.62	81.19
+ Nucleus Sampling	73.34	84.95	68.94	81.72	70.01	82.51	70.19	84.72	63.93	79.54	65.45	80.80
+ QPP (Ours)	**74.68**	**85.92**	**70.05**	**83.47**	**71.10**	**84.06**	**70.83**	**85.76**	**65.33**	**80.64**	**66.67**	**81.88**
BERT-SQG [34]	70.19	81.47	66.05	79.64	67.05	80.08	65.06	82.20	59.59	78.04	60.92	79.05
+ BeamSearch	73.71	84.44	68.71	81.98	69.93	82.58	67.31	82.54	61.94	79.02	63.25	79.88
+ Top-k Sampling	72.81	84.16	69.20	82.24	70.07	82.71	69.12	84.20	60.44	78.27	62.55	79.71
+ Nucleus Sampling	70.73	83.60	68.56	81.80	69.09	82.24	67.74	83.16	61.61	78.74	63.09	79.81
+ QPP (Ours)	**74.36**	**85.53**	**70.77**	**83.60**	**71.64**	**84.07**	**69.23**	**85.38**	**64.21**	**80.53**	**65.43**	**81.71**

III by sampling generated questions from QG models and their variants and is used to tune the hyper-parameters. In the following sections, we consider emrQA as the *source* dataset and our annotated MIMIC-III QA dataset as the *target* data. Detailed statistics of the two datasets are in Table I.

V. Experimental Setup

A. Base QG models

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

- **NQG** [10] is the first seq2seq model with a global attention mechanism [32] for question generation.
- **NQG++** [15] is one of the most commonly adopted QG baselines with a feature-enriched encoder (e.g., lexical features) and a copy mechanism [35].
- **BERT-SQG** [34] uses a pretrained BERT model (we use ClinicalBERT [31] to accommodate clinical setting) as the encoder and formulates the decoding as a "MASK" token prediction problem.

It has been studied that beam search and sampling strategies show competitive performance in diversifying generations [36], [37]. We thus include Top-k [38] and Nucleus samplings [39] as representative sampling strategies in our experiments.

As such, to investigate the effectiveness of diverse QG for QA, we consider the following variants of each base QG model: (1) Base Model: Inference with greedy search; (2) Base Model + Beam Search: Inference with Beam Search of beam size K and keep top K beams ($K = 3$); (3) Base Model + Top-k sampling: Inference with sampling from top-k tokens ($k = 20$); (4) Base Model + Nucleus sampling: Inference with sampling from top-p tokens ($p = 0.95$); (5) Base Model + QPP: Inference with greedy search for both QPP module and Base model.

B. Base QA models

For QA, we instantiate `CliniQG4QA` with two base models, DocReader [6] and ClinicalBERT [31]. When training a QA model, we only use the synthetic data on the target contexts and do not combine the synthetic data with the source data since the combination does not help in our preliminary studies.

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.

C. Evaluation Metrics

For QA evaluation, we report exact match (EM) (percentage of predictions that match the ground truth answers exactly) and F1 (average overlap between the predictions and ground truth answers) as in [40]. Since our main goal is to evaluate whether the generated questions are useful to improve the QA performance on the target contexts, the common language generation metrics such as BLEU [41] and ROUGE-L [42] are not suitable to reflect the quality of the generated questions, and thus we do not adopt these metrics in our experiments.

D. Implementation Details

Base QG Models: We re-implement three base QG models using Pytorch and have ensured that they achieve comparable performance as originally reported. Best QG models are selected using the per-token accuracy of both the QPP module (if applicable) and QG on dev set.

Base QA Models: We use the open-sourced implementation.[6] Best QA models are selected using EM and F1 on dev set.

Hyperparameters Search: Hyperparameters of QG models are set to be the same as in original papers and hyperparameters of QA models are set according to [8]. Specifically, we train NQG and NQG++ up to 20 epochs, BERT-SQG up to 5 epochs, DocReader up to 5 epochs and ClinicalBERT up to 3 epochs.

VI. Experimental Results

A. Can Generated Questions Help QA on New Contexts?

Table II summarizes the performance of two widely used QA models, DocReader [6] and ClinicalBERT [31], 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 and sampling strategies could boost the diversity of generated questions to some extent, and thus lead to the improvement of QA models, our proposed QPP module can improve the QA performance by a larger margin. 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 downstream QA on new contexts, and our simple QPP module can help existing QG models achieve such a goal.

B. 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 Fig 4, 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.

A Closer Look at Generated Question Types. To further demonstrate QPP module can help generate diverse questions, we show the distribution over the types of questions generated by NQG-based models in Fig 3.

We observe that questions generated by base NQG and NQG+BeamSearch are limited in terms of the question types. However, more types of questions (e.g., “How”, “Why”) can be generated when enabling sampling strategies. Furthermore, when being equipped with our QPP module, the NQG model can even generate questions of an extremely rare type, i.e., ”When” questions. Though Top-k and Nucleus sampling methods also generate questions of less frequent types, our QPP module could cover even more types.

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

C. Diverse Questions Really Matter for QA: Two Real Cases.

In Fig 5, we present a QA example and a QG example from MIMIC-III for qualitative analysis.

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, as shown in Fig 3, the NQG model and “NQG+BeamSearch” cannot generate any “why” questions and sampling strategies could only help generate a limited number of “why” questions. 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 as well.

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. Sampling strategies, though further diversifying the generation during decoding, suffer from generating irrelevant contents (e.g., “NQG+Nucleus” generates a irrelevant “morphine” token). Enabling our QPP module helps generate relevant and diverse questions including “Why”, “What”, “How”, etc.

D. Ablation Study

Performance of QPP with Sampling Strategies. Since our QPP is compatible with sampling strategies, we further study the performance after combining these two techniques. Table III shows the results, which indicate that combining two techniques can improve the sampling strategies' performance but do not

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

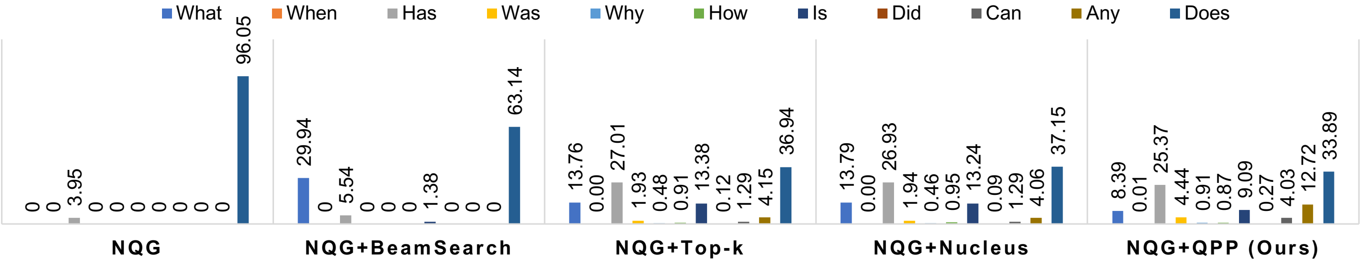

Fig. 3: Distributions over types of questions generated by NQG models.

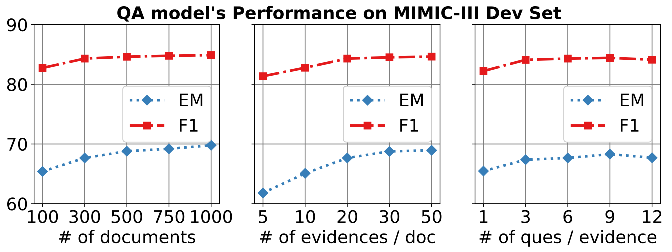

Fig. 4: Influence of the number of documents, number of evidences per document, number of QA pairs per evidence on QA performance.

TABLE III: The QA performance on MIMIC-III test set when QPP is employed with sampling strategies

QA Datasets	DocReader [6]					
	Human Generated		Human Verified		Overall Test	
	EM	F1	EM	F1	EM	F1
NQG	66.99	79.67	64.71	79.36	65.26	79.43
+ QPP	**74.36**	**85.18**	**68.82**	**82.89**	**70.09**	**83.44**
+ Top-k	71.58	83.48	66.77	80.45	67.94	81.19
+ Tok-k + QPP	72.52	84.98	67.67	81.79	68.84	82.56
+ Nucleus	70.62	83.68	67.16	80.37	68.00	81.17
+ Nucleus + QPP	74.12	85.08	68.10	81.36	69.56	82.26

lead to further improvement compared with using QPP only. This demonstrate that our QPP module is good enough to generate diverse useful questions for improving QA.

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 methods: binary relevance (BR) and classifier chain (CC) [43], [44]. BR develops multiple binary classifiers independently while CC builds a chain of classifiers and predicts labels sequentially. We use multi-layer perceptron as the base model for both BR and CC. For each answer evidence, the input is the representation from the same LSTM encoder as our QPP module.

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: 5. abd pain:
-NQG+BeamSearch: 5. abd pain:
-NQG+Top-k: 5. abd pain:
-NQG+Nucleus: 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+Top-k: **Has** the patient ever had any pain? **Has** the patient ever reported pain? **Does** the patient have a history pain?
-NQG+Nucleus: **Has** the patient ever gone into pain? **What** happened when she was given morphine? Is there mention pain anywhere in the record?
-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? ...

Fig. 5: QA and QG examples. The red parts in contexts are ground-truth answer evidences.

TABLE IV: Choosing seq2seq-based QPP over alternative multi-label classification methods. HL: Hamming Loss.

Models	HL	Precision	Recall	F1
Binary Relevance	0.0524	**99.22**	90.89	94.87
Classifier Chain	0.0524	**99.22**	90.89	94.87
QPP	**0.0346**	97.28	**96.20**	**96.74**

From Table IV, we can see: (1) The seq2seq design in our 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.

VII. Conclusion

This paper proposes a simple yet effective framework for improving clinical QA on new contexts. It leverages a seq2seq-based question phrase prediction 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.

Acknowledgment

The authors would like to thank all the constructive reviews. The research is sponsored in part by the PCORI 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 [45]. 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

[1] J. Patrick and M. Li, “An ontology for clinical questions about the contents of patient notes,” *JBI*, vol. 45, no. 2, pp. 292–306, 2012.

[2] P. Raghavan, S. Patwardhan, J. J. Liang, and M. V. Devarakonda, “Annotating electronic medical records for question answering,” *arXiv preprint arXiv:1805.06816*, 2018.

[3] A. Pampari, P. Raghavan, J. Liang, and J. Peng, “emrqa: A large corpus for question answering on electronic medical records,” in *EMNLP’18*, 2018, pp. 2357–2368.

[4] J. Fan, “Annotating and characterizing clinical sentences with explicit why-qa cues,” in *NAACL Clinical NLP Workshop*, 2019, pp. 101–106.

[5] B. P. S. Rawat, W.-H. Weng, P. Raghavan, and P. Szolovits, “Entity-enriched neural models for clinical question answering,” *arXiv preprint arXiv:2005.06587*, 2020.

[6] D. Chen, A. Fisch, J. Weston, and A. Bordes, “Reading wikipedia to answer open-domain questions,” in *ACL’17*, 2017, pp. 1870–1879.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” in *NAACL-HLT’19*, 2019, pp. 4171–4186.

[8] X. Yue, B. J. Gutierrez, and H. Sun, “Clinical reading comprehension: A thorough analysis of the emrqa dataset,” in *ACL’20*, 2020.

[9] A. E. Johnson, T. J. Pollard, L. Shen, H. L. Li-wei, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark, “Mimic-iii, a freely accessible critical care database,” *Scientific data*, vol. 3, p. 160035, 2016.

[10] X. Du, J. Shao, and C. Cardie, “Learning to ask: Neural question generation for reading comprehension,” in *ACL’17*, 2017, pp. 1342–1352.

[11] M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi, “Bidirectional attention flow for machine comprehension,” in *ICLR’17*, 2017.

[12] A. Wen, M. Y. Elwazir, S. Moon, and J. Fan, “Adapting and evaluating a deep learning language model for clinical why-question answering,” *JAMIA Open*, vol. 3, no. 1, pp. 16–20, 2020.

[13] A. Poliak, M. Fleming, C. Costello, K. W. Murray, M. Yarmohammadi, S. Pandya, D. Irani, M. Agarwal, U. Sharma, S. Sun, N. Ivanov, L. Shang, K. Srinivasan, S. Lee, X. Han, S. Agarwal, and J. Sedoc, “Collecting verified COVID-19 question answer pairs,” in *Proceedings of the 1st Workshop on NLP for COVID-19@ EMNLP 2020, Online, December 2020*. Association for Computational Linguistics, 2020.

[14] X. F. Zhang, H. Sun, X. Yue, S. M. Lin, and H. Sun, “COUGH: A challenge dataset and models for COVID-19 FAQ retrieval,” in *EMNLP 2021*. Association for Computational Linguistics, 2021, pp. 3759–3769.

[15] Q. Zhou, N. Yang, F. Wei, C. Tan, H. Bao, and M. Zhou, “Neural question generation from text: A preliminary study,” in *NLPCC’17*. Springer, 2017, pp. 662–671.

[16] X. Sun, J. Liu, Y. Lyu, W. He, Y. Ma, and S. Wang, “Answer-focused and position-aware neural question generation,” in *EMNLP’18*, 2018, pp. 3930–3939.

[17] Y. Zhao, X. Ni, Y. Ding, and Q. Ke, “Paragraph-level neural question generation with maxout pointer and gated self-attention networks,” in *EMNLP’18*, 2018, pp. 3901–3910.

[18] P. Nema, A. K. Mohankumar, M. M. Khapra, B. V. Srinivasan, and B. Ravindran, “Let’s ask again: Refine network for automatic question generation,” in *EMNLP-IJCNLP’19*, 2019, pp. 3305–3314.

[19] L. A. Tuan, D. J. Shah, and R. Barzilay, “Capturing greater context for question generation,” in *AAAI’20*, 2020.

[20] Z. Yang, J. Hu, R. Salakhutdinov, and W. Cohen, “Semi-supervised qa with generative domain-adaptive nets,” in *ACL’17*, 2017, pp. 1040–1050.

[21] X. Du and C. Cardie, “Harvesting paragraph-level question-answer pairs from wikipedia,” in *ACL’18*, 2018, pp. 1907–1917.

[22] C. Alberti, D. Andor, E. Pitler, J. Devlin, and M. Collins, “Synthetic qa corpora generation with roundtrip consistency,” in *ACL’19*, 2019, pp. 6168–6173.

[23] S. Zhang and M. Bansal, “Addressing semantic drift in question generation for semi-supervised question answering,” in *EMNLP-IJCNLP’19*, 2019, pp. 2495–2509.

[24] J. Kang, H. P. San Roman *et al.*, “Let me know what to ask: Interrogative-word-aware question generation,” in *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, 2019, pp. 163–171.

[25] J. Cho, M. Seo, and H. Hajishirzi, “Mixture content selection for diverse sequence generation,” in *EMNLP-IJCNLP’19*, 2019, pp. 3112–3122.

[26] B. Liu, H. Wei, D. Niu, H. Chen, and Y. He, “Asking questions the human way: Scalable question-answer generation from text corpus,” in *WWW’20*, 2020, pp. 2032–2043.

[27] S. Shen, Y. Li, N. Du, X. Wu, Y. Xie, S. Ge, T. Yang, K. Wang, X. Liang, and W. Fan, “On the generation of medical question-answer pairs.” in *AAAI*, 2020, pp. 8822–8829.

[28] S. Soni and K. Roberts, “A paraphrase generation system for ehr question answering,” in *18th BioNLP Workshop*, 2019, pp. 20–29.

[29] ——, “Paraphrasing to improve the performance of electronic health records question answering,” *AMIA Summits*, vol. 2020, p. 626, 2020.

[30] L. A. Ramshaw and M. P. Marcus, “Text chunking using transformation-based learning,” in *Natural language processing using very large corpora*. Springer, 1999, pp. 157–176.

[31] E. Alsentzer, J. R. Murphy, W. Boag, W.-H. Weng, D. Jin, T. Naumann, and M. McDermott, “Publicly available clinical bert embeddings,” *NAACL Clinical NLP Workshop 2019*, 2019.

[32] M.-T. Luong, H. Pham, and C. D. Manning, “Effective approaches to attention-based neural machine translation,” in *EMNLP’15*, 2015, pp. 1412–1421.

[33] n2c2, “n2c2 nlp research data sets,” portal.dbmi.hms.harvard.edu, 2006. [Online]. Available: https://portal.dbmi.hms.harvard.edu/projects/n2c2-nlp/

[34] Y.-H. Chan and Y.-C. Fan, “A recurrent bert-based model for question generation,” in *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, 2019, pp. 154–162.

[35] C. Gulcehre, S. Ahn, R. Nallapati, B. Zhou, and Y. Bengio, “Pointing the unknown words,” in *ACL’16*, 2016, pp. 140–149.

[36] D. Ippolito, R. Kriz, J. Sedoc, M. Kustikova, and C. Callison-Burch, “Comparison of diverse decoding methods from conditional language models,” in *ACL ’19*. Association for Computational Linguistics, 2019, pp. 3752–3762.

[37] M. A. Sultan, S. Chandel, R. F. Astudillo, and V. Castelli, “On the importance of diversity in question generation for QA,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020*, D. Jurafsky, J. Chai, N. Schluter, and J. R. Tetreault, Eds., 2020, pp. 5651–5656.

[38] A. Fan, M. Lewis, and Y. N. Dauphin, “Hierarchical neural story generation,” in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018*, 2018, pp. 889–898.

[39] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, “The curious case of neural text degeneration,” in *International Conference on Learning Representations*, 2019.

[40] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, “Squad: 100,000+ questions for machine comprehension of text,” in *EMNLP’16*, 2016, pp. 2383–2392.

[41] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “Bleu: a method for automatic evaluation of machine translation,” in *ACL’02*. Association for Computational Linguistics, 2002, pp. 311–318.

[42] C.-Y. Lin, “ROUGE: A package for automatic evaluation of summaries,” in *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, Jul. 2004, pp. 74–81.

[43] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, “Learning multi-label scene classification,” *Pattern Recognition*, p. 1757–1771, 2004.

[44] J. Read, B. Pfahringer, G. Holmes, and E. Frank, “Classifier chains for multi-label classification,” *Machine learning*, vol. 85, no. 3, p. 333, 2011.

[45] O. S. Center, “Ohio supercomputer center,” 1987. [Online]. Available: http://osc.edu/ark:/19495/f5s1ph73