# Leveraging Context for Neural Question Generation in Open-domain Dialogue Systems

Yanxiang Ling
Science and Technology on Information Systems Engineering Laboratory
National University of Defense Technology
lingyanxiang@nudt.edu.cn

Fei Cai*
Science and Technology on Information Systems Engineering Laboratory
National University of Defense Technology
caifei@nudt.edu.cn

Honghui Chen
Science and Technology on Information Systems Engineering Laboratory
National University of Defense Technology
chenhonghui@nudt.edu.cn

Maarten de Rijke
Informatics Institute
University of Amsterdam
derijke@uva.nl

## ABSTRACT

Question generation in open-domain dialogue systems is a challenging but less-explored task. It aims to enhance the interactivity and persistence of human-machine interactions. Previous work mainly focuses on question generation in the setting of single-turn dialogues, or investigates it as a data augmentation method for machine comprehension. We propose a Context-augmented Neural Question Generation (CNQG) model that leverages the conversational context to generate questions for promoting interactivity and persistence of multi-turn dialogues. More specifically, we formulate the task of question generation as a two-stage process. First, we employ an encoder-decoder framework to predict a question pattern, which denotes a set of representative interrogatives, and identify the potential topics from the conversational context by employing point-wise mutual information. Then, we generate the question by decoding the concatenation of the current dialogue utterance, the pattern, and the topics with an attention mechanism. To the best of our knowledge, ours is the first work on question generation in multi-turn open-domain dialogue systems. Our experimental results on two publicly available multi-turn conversation datasets show that CNQG outperforms the state-of-the-art baselines in terms of BLEU-1, BLEU-2, Distinct-1 and Distinct-2. In addition, we find that CNQG allows one to efficiently distill useful features from long contexts, and maintain robust effectiveness even for short contexts.

## CCS CONCEPTS

• **Information systems → Users and interactive retrieval**.

## KEYWORDS

Question generation, dialogue systems, context, open-domain.

*Corresponding author.


*WWW '20, April 20–24, 2020, Taipei, Taiwan*

ACM ISBN 978-1-4503-7023-3/20/04.
https://doi.org/10.1145/3366423.3379996

**ACM Reference Format:**
Yanxiang Ling, Fei Cai, Honghui Chen, and Maarten de Rijke. 2020. Leveraging Context for Neural Question Generation in Open-domain Dialogue Systems. In *Proceedings of The Web Conference 2020 (WWW '20), April 20–24, 2020, Taipei, Taiwan.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3366423.3379996

## 1 INTRODUCTION

Question Generation (QG) aims to generate a relevant question for a given input. It has been used to automatically create large-scale training data for machine reading comprehension [20] and question answering [17, 22]. In the field of open-domain dialogue systems, question generation, also known as learning to ask, serves as an essential communication skill to help solicit feedback from users and to extend current conversational topics or start new ones, which can enhance the interactivity and persistence of dialogues [26].

Most previous work on question generation uses neural methods that adopt a sequence-to-sequence framework (Seq2Seq, also called encoder-decoder framework) [10, 21]. For instance, Serban et al. [17] apply a Seq2Seq framework to generate factoid questions from a structured knowledge base. Du et al. [4] change the modality of the input data and generate questions based on given text passages and answers, which has inspired follow-up work that includes [5, 20, 22]. In addition, Mostafazadeh et al. [14] focus on the novel task of visual question generation (VQG) that involves generating a natural question for a given image. However, QG in open-domain dialogue systems is still challenging. First, the main purpose of QG is to achieve interactive and persistent dialogues [26], which is substantially different from the traditional QG tasks, where questions are generated to enhance machine comprehension and usually can be answered by the given input. In addition, colloquial and short texts in conversational corpora are often creative in the expressions they use and semantically ambiguous, which increases the difficulty of QG. For instance, a phrase like "*I don't know*" frequently occurs in dialogues [6], which often has a negative impact on the informativeness and diversity of generated questions [11, 27].

To address the above issue, Li et al. [12] explore how a chatbot can return an appropriate answer by asking questions in a movie-specific domain. However, this model resorts to a specific knowledge base, which restricts their solution to dialogues in a

closed domain. Wang et al. [26] focus on QG in the setting of open-domain dialogue systems and propose a typed decoder based on a Seq2Seq architecture, which takes the latest dialogue utterance as input. Their solution ignores the previous conversational context, which limits its application to single-turn dialogues. For multi-turn conversations, generating a random or free-style question without considering its conversational context is not useful for enhancing interactivity and persistence of the dialogue. Hence, we argue that a good question should contain relevant topics that may appear in previous utterances [24].

In this paper, we investigate the task of Question Generation (QG) in the setting of multi-turn open-domain dialogue systems and propose a *Context-augmented Neural Question Generation* (CNQG) model, that leverages the conversational context in dialogues to generate appropriate and informative questions. In particular, we formulate the question generation task as a two-stage process that is implemented within an encoder-decoder framework: (1) We first encode the latest dialogue utterance, referred to as the "post," into a hidden vector representation, which is then used to predict the question *pattern* that denotes a set of representative interrogatives. We then use point-wise mutual information to identify the *topics* from the preceding conversational context as well as the post. (2) We employ an encoder-decoder framework with an attention mechanism to generate the final question by decoding the concatenation of the post, the pattern and the topics as input. To evaluate the performance of CNQG, we conduct experiments on two publicly available benchmark datasets, i.e., the DailyDialog dataset[1] and the Cornell Movie-Dialog dataset[2] ("Cornell" for short), which are both collections of multi-turn dialogues extracted from human-to-human conversations. Experimental results show that CNQG outperforms the state-of-the-art baselines in terms of BLEU-1, BLEU-2, Distinct-1 and Distinct-2, which demonstrates its effectiveness at generating appropriate and informative questions. In addition, we find that CNQG can efficiently avoid interference from long contexts so as to prevent digressions, and maintain robust effectiveness even for short contexts, which are usually ambiguous.

The main contributions of our work are the following.

- To the best of our knowledge, ours is the first to work on question generation in multi-turn open-domain dialogue systems. We leverage the conversational context to generate appropriate and informative question.
- We propose a context-augmented neural question generation model (CNQG) that models question generation as a two-stage process and follows an encoder-decoder framework to generate questions.
- We analyze the effectiveness of CNQG on two conversational datasets and find that it significantly beats the state-of-the-art baselines in terms of BLEU-1 and BLEU-2.

## 2 APPROACH

We provide a high-level overview of the Context-augmented Neural Question Generation (CNQG) model in Fig. 1. CNQG consists of four main components, i.e., a post encoder (§2.1), a pattern predictor (§2.2), a topic identifier (§2.3), and a question generator (§2.4).

We first detail the task of question generation in multi-turn open-domain dialogue systems. We take a $d$-turn ($d \geq 3$) dialogue session as a sequence $\{U_1, \ldots, U_d\}$, which is then represented by a triple $(C, X, Y)$, where $C$ denotes the *conversational context* consisting of $d-2$ utterances $\{U_1, \ldots, U_{d-2}\}$, $X$ is the *post* $U_{d-1}$, and $Y$ represents the target *question* $U_d$. The purpose of question generation in multi-turn open-domain dialogue systems is to compute the probability $P(Y \mid X, C)$ of generating a question $Y$ given the conversational context $C$ and post $X$.

We assume that the target question $Y$ consists of a sequence of $T$ words, i.e., $Y = (y_1, \ldots, y_T)$, and is an implicit combination of a question pattern $Z$ and topics $K$. The question pattern $Z = (z_1, \ldots, z_L)$ comprises $L$ representative interrogatives. The topics $K = \{k_1, \ldots, k_M\}$ are a set of words semantically related to the post and the conversational context. Thus, question generation can be regarded as a two-stage process. First, use the post to predict the question pattern $P(Z \mid X)$ and leverage the post plus the context to obtain the question topics $P(K \mid X, C)$. Second, decode the concatenation of the post, pattern and topics to generate the final question word-by-word as

$$P(y_1, \ldots, y_T) = \prod_{t=1}^{T} P(y_t \mid X, Z, K, y_{<t}), \tag{1}$$

where $y_t$ is the word to be generated at the $t$-th step, and $y_{<t}$ represents the previously generated words before the $t$-th step.

### 2.1 Post encoding

Given an $N$-length post $X = (x_1, \ldots, x_N)$, we use a GRU-based encoder to convert the post sentence into a sequence of hidden vectors as:

$$\mathbf{h}^X_{n+1} = \text{GRU}(\mathbf{h}^X_n, \mathbf{e}_{x_{n+1}}), \tag{2}$$

where $0 \leq n < N$ and $\mathbf{e}_{x_{n+1}}$ is the embedding of word $x_{n+1}$. The GRU is parameterized as follows:

$$\begin{aligned} \mathbf{z} &= \sigma_g(\mathbf{W_z}\mathbf{x}_{n+1} + \mathbf{U_z}\mathbf{h}^X_n) \\ \mathbf{r} &= \sigma_g(\mathbf{W_r}\mathbf{x}_{n+1} + \mathbf{U_r}\mathbf{h}^X_n) \\ \mathbf{s} &= \sigma_h(\mathbf{W_s}\mathbf{x}_{n+1} + \mathbf{U_s}(\mathbf{h}^X_n \circ \mathbf{r})) \\ \mathbf{h}^X_{n+1} &= (1 - \mathbf{z}) \circ \mathbf{s} + \mathbf{z} \circ \mathbf{h}^X_n, \end{aligned} \tag{3}$$

where $\mathbf{x}_{n+1}$ is the input vector and is assigned as $\mathbf{e}_{x_{n+1}}$ here; $\mathbf{z}$ and $\mathbf{r}$ are the update gate vector and reset gate vector, respectively; $\mathbf{W_z}$, $\mathbf{U_z}$, $\mathbf{W_r}$, $\mathbf{U_r}$, $\mathbf{W_s}$, $\mathbf{U_s}$ are the weight matrices; $\circ$ represents the operation of element-wise multiplication, $\sigma_g$ and $\sigma_h$ are the activation functions.

We convert a post $X$ into a sequence of hidden states ($\mathbf{h}^X_1, \ldots, \mathbf{h}^X_N$), which is fed to the following decoders for pattern prediction and question generation, respectively.

### 2.2 Pattern prediction

Most naturally occurring questions in human conversations feature one of a small set of interrogatives [5]. For instance, a question "*What is your nationality?*" features the interrogative *what*. Following [5], we identify 8 types of question pattern: *yes/no*, *what*, *why*, *how*, *who*, *where*, *when* and *which*. Each pattern is expressed by one or several interrogatives, e.g., the pattern *who* has the interrogatives *who*, *whose*, *whom*.

[1] The dataset is available at http://yanran.li/dailydialog

[2] The dataset is available at http://www.cs.cornell.edu/~cristian/Cornell_Movie-Dialogs_Corpus.html.

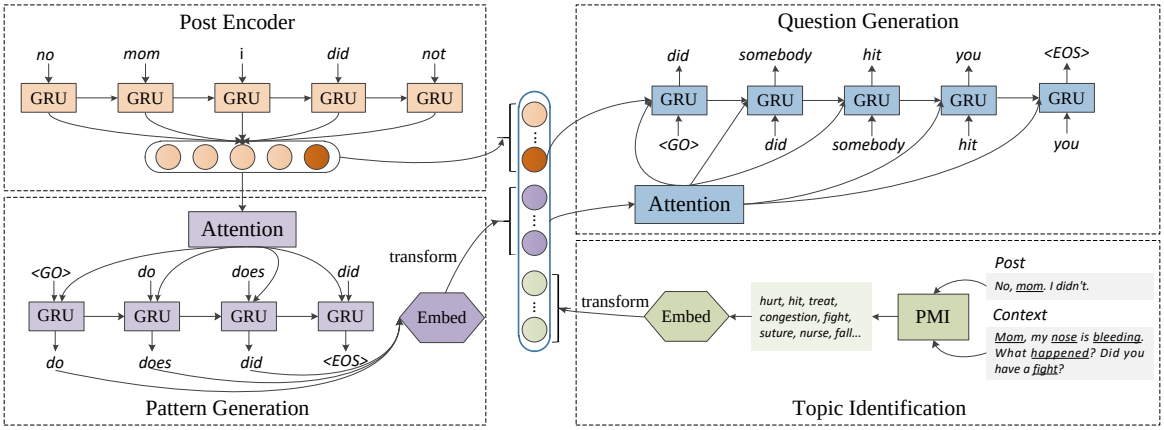



**Figure 1: Overview of the Context-augmented Neural Question Generation (CNQG) framework.**

We first collect commonly used interrogatives to construct a pattern vocabulary, then adopt an attention-augmented encoder-decoder framework to generate question pattern-related interrogatives $Z = (z_1, \ldots, z_L)$, as follows:

$$P(Z \mid X) = \prod_{l=1}^{L} P(z_l \mid X, z_{<l}). \tag{4}$$

The word probability distribution at each decoding position is computed as follows:

$$P(z_l = v \mid X, z_{<l}) = \frac{\exp(g(\mathbf{s}_l^Z, v))}{\sum_{v' \in V^Z} \exp(g(\mathbf{s}_l^Z, v'))}, \tag{5}$$

where $v$ is a word from the pattern vocabulary $V^Z$; $g$ is the projection function implemented by a fully-connected layer with a softmax function; $\mathbf{s}_l^Z$ is the pattern decoder hidden state at $l$-step and is computed as:

$$\mathbf{s}_0^Z = \mathbf{h}_N^X \tag{6}$$

$$\mathbf{s}_l^Z = \text{GRU}(\mathbf{s}_{l-1}^Z, [\mathbf{c}_l^Z, \mathbf{e}_{z_{l-1}}]), \tag{7}$$

where the GRU for pattern prediction is similar to Eq. 3 but has different parameters; $[\mathbf{c}_l^Z, \mathbf{e}_{z_{l-1}}]$ denotes a concatenation of $\mathbf{e}_{z_{l-1}}$ and $\mathbf{c}_l^Z$, where $\mathbf{e}_{z_{l-1}}$ is the embedding of word $z_{l-1}$; $\mathbf{c}_l^Z$ is a weighted mixture vector computed by attentively reading the output (see Eq. 2) of the post encoder as:

$$\mathbf{c}_l^Z = \sum_{n=1}^{N} \alpha_{ln}^Z \mathbf{h}_n^X, \tag{8}$$

where the weight $\alpha_{ln}^Z$ is defined by

$$\alpha_{ln}^Z = \frac{\exp(e_{ln})}{\sum_{j=1}^{N} \exp(e_{lj})}, \quad e_{ln} = \eta(\mathbf{s}_{l-1}^Z, \mathbf{h}_n^X). \tag{9}$$

Here, $\eta$ is implemented by a multi-layer perceptron model with tanh as the activation function.

## 2.3 Context-augmented topic identification

To maintain consistency with previous utterances in a given dialogue, we propose a topic identification scheme to find the potential topics in the conversational context as well as the post. We first locate the nouns and verbs from the context as well as the post, and then identify their topics using point-wise mutual information (PMI) [2] matrices. PMI is often used to measure similarity between two items and previous studies [7, 25, 26] have shown its effectiveness in natural language processing.

Instead of using the traditional PMI, we introduce a part-of-speech (POS) feature to guide the computation process and obtain two POS-based PMI matrices, i.e., one corresponding to the noun-type PMI and the other corresponding to the verb-type PMI. More specifically, we first apply POS tagging to identify nouns and verbs in the context as well as in the post; we refer to nouns and verbs in the context and post as *triggers*, and to those in the ground-truth questions as *targets*. Then, the PMI scores of pairs of trigger and target nouns and of pairs of trigger and target verbs are calculated as follows:

$$PMI(w_1, w_2) = \log \frac{p_{\langle trigger, target \rangle}(w_1, w_2)}{p_{trigger}(w_1) \cdot p_{target}(w_2)}, \tag{10}$$

where $p_{\langle trigger, target \rangle}(w_1, w_2)$ is the co-occurrence probability of $w_1$ occurring in triggers and $w_2$ occurring in targets, simultaneously; $p_{trigger}(w_1)$ and $p_{target}(w_2)$ denote the independent probabilities of $w_1$ occurring as a trigger and $w_2$ as a target, respectively.

Given a post $X$ and a conversational context $C$, we determine the relevance score for a word $k_m$ (a "topic") as a sum of its PMI scores:

$$P(k_m \mid X, C) = \sum_{i=1}^{N_{noun}} PMI_{noun}(w_i, k_m) + \sum_{j=1}^{N_{verb}} PMI_{verb}(w_j, k_m), \tag{11}$$

where $w_i$ ranges over nouns from the $N_{noun}$-length noun set extracted from the post and the context and $w_j$ is a verb from the $N_{verb}$-length verb set. Finally, we select the top-$M$ words $k_1, \ldots, k_M$ with the highest relevance scores as the topics for the given post $X$ and context $C$.

## 2.4 Question generation

The question decoder is similar to the pattern decoder and takes a vector as input and generates the question word-by-word with an attention mechanism. Here, the input to the question decoder is a concatenated vector $\mathbf{\Psi}$ of three sources, namely the post, the pattern and the topics, which is obtained as follows:

$$\mathbf{\Psi} = [\mathbf{h}_1^X, \ldots, \mathbf{h}_N^X, \mathbf{p}_{z_1}, \ldots, \mathbf{p}_{z_L}, \mathbf{t}_{k_l}, \ldots, \mathbf{t}_{k_M}], \tag{12}$$

$$\mathbf{p}_{z_l} = \mathbf{W}\mathbf{e}_{z_l}, \quad \mathbf{t}_{k_m} = \mathbf{W}\mathbf{e}_{k_m}, \tag{13}$$

where $\mathbf{p}_{z_l}$ and $\mathbf{t}_{k_m}$ are the transformed vectors; $\mathbf{e}_{z_l}$ and $\mathbf{e}_{k_m}$ are the embeddings of the generated interrogative $z_l$ (see §2.2) and identified topic $k_m$ (see §2.3), respectively; $\mathbf{W} \in \mathbb{R}^{d_h \times d_e}$ is used to transform the embedding vectors (e.g., $\mathbf{e}_{z_l}$ and $\mathbf{e}_{k_m}$).

Given the concatenated vector $\mathbf{\Psi}$, the GRU for question generation has similar structure with Eq. 3, but is assigned as:

$$\mathbf{s}_0^Y = \mathbf{h}_N^X \tag{14}$$

$$\mathbf{s}_t^Y = \text{GRU}(\mathbf{s}_{t-1}^Y, [\mathbf{c}_t^Y, \mathbf{e}_{y_{t-1}}]), \tag{15}$$

where $[\mathbf{c}_t^Y, \mathbf{e}_{y_{t-1}}]$ is the concatenation of $\mathbf{e}_{y_{t-1}}$ and $\mathbf{c}_t^Y$. $\mathbf{e}_{y_{t-1}}$ is the embedding of generated word at step $t-1$, $\mathbf{c}_t^Y$ is a weighted sum vector obtained from the attention mechanism as follows:

$$\mathbf{c}_t^Y = \sum_{i=1}^{(N+L+M)} \alpha_{ti}^Y \varphi_i, \tag{16}$$

where $\varphi_i \in \mathbf{\Psi}$ and the weight coefficient $\alpha_{ti}^Y$ is computed as

$$\alpha_{ti}^Y = \frac{\exp(e_{ti})}{\sum_{i=1}^{(N+L+M)} \exp(e_{ti})}, \quad e_{ti} = \eta(\mathbf{s}_{t-1}^Y, \varphi_i). \tag{17}$$

Here, $\eta$ is defined similarly with the question pattern decoder. The probability $P(y_t \mid X, Z, K, y_{<t})$ of word $y_t$ is obtained as follows:

$$P(y_t = w \mid X, Z, K, y_{<t}) = \frac{\exp(g(\mathbf{s}_t^Y, w))}{\sum_{w' \in V^Y} \exp(g(\mathbf{s}_t^Y, w'))}, \tag{18}$$

where $w$ is a word from the pre-defined vocabulary $V^Y$.

In the training phase, the proposed model is trained by minimizing the negative log-likelihood of the training question $Y$, where the loss function $L_\theta = L_\theta(Z) + L_\theta(Y)$ has two components:

$$\begin{aligned} L_\theta(Z) &= -\sum_{l=1}^{L} \log P(z_l \mid X, z_{<l}) \\ L_\theta(Y) &= -\sum_{t=1}^{T} \log P(y_t \mid X, Z, K, y_{<t}), \end{aligned} \tag{19}$$

where $L_\theta(Z)$ and $L_\theta(Y)$ are losses from the pattern decoder and the question decoder, respectively; $\theta$ denotes the parameter set. Here, $L_\theta(Z)$ provides an additional supervised signal for pattern prediction.

# 3 EXPERIMENTAL SETUP

In this section, we detail our experimental setup. We focus on three research questions. (RQ1) Does CNQG outperform competitive baselines on question generation? (RQ2) How does CNQG perform on predicting question patterns? (RQ3) What is the impact of context length in our model on question generation?

## 3.1 Datasets

We conduct experiments on two multi-turn conversational datasets, i.e., the DailyDialog dataset [13] and the Cornell Movie-Dialog dataset [3]. DailyDialog is collected from human-to-human talks in daily life. It contains 11,318 human-written dialog sessions and covers various topics such as culture, education, tourism and health etc. Cornell is extracted from movie scripts including 220,579 conversational exchanges between 10,292 pairs of movie characters.

To train CNQG, we perform several pre-processing steps on the raw text. We first generate *triples* $(C, X, Y)$, i.e., three turn dialogues between two interlocutors where $C$ is the context, $X$ is the post, and $Y$ is the target response. Then, with the help of hand-crafted rules we pick triples where the response is in the form of a question. These rules include presence of a question mark and a list of interrogatives. We identify the pattern for each question based on the classification method proposed in [5]. Finally, we obtain 28,769 triples from DailyDialog and 49,689 triples from Cornell; for each dataset, 2,000 triples are randomly selected for validation and another 2,000 for testing; the remainder is used for training. The statistics of the datasets we use are shown in Table 1. Clearly, the distributions of different patterns are quite unbalanced. Moreover, both datasets feature a broad range of context lengths.

## 3.2 Baselines and metrics

*3.2.1 Baselines.* For comparison, we compare the performance of CNQG against three state-of-the-art baselines for question generation: (1) **NQG** [4]: an attention-based sequence to sequence learning model that encodes sentences from a text passage to generate a question. Similar approaches can be found in [20, 26]. Here, we set the post as the input sentence. (2) **DCGM-I** [19]: a context-sensitive generation model in dialogues, where the context and the post are encoded into a fixed-length vector that is used to generate responses. (3) **HRED** [18]: a hierarchical encoder-decoder model that introduces an additional context encoder to model the interactive structure of multi-turn dialogues.

*3.2.2 Metrics.* Following [4, 5, 17, 20, 26], we adopt five metrics to evaluate the performance of CNQG and the baselines, i.e., BLEU-1 [15], BLEU-2 [15], Distinct-1 [11], Distinct-2 [11]. BLEU-1 and BLEU-2 are the most frequently used metrics for question generation; they measure the word-overlap between the generated question and the ground-truth. A higher BLEU score indicates that the generated question is closer to the ground-truth. Distinct-1 and Distinct-2 respectively evaluate the number of distinct unigrams and bigrams in the generated questions, which are often used to measure the questions in terms of sentence diversity.

## 3.3 Implementation details

In our experiments, we manually collect 36 interrogatives as the pattern vocabulary. We adopt the NLTK tool[3] for pos-tagging and lemmatization. In total, 30 topics are identified for each dialogue. Like [4, 20], the word embedding is initialized by pre-trained Glove 6B[4] word vectors with 300 dimensions. We use the original vocabulary consisting of 16,578 unique words in DailyDialog for decoding and choose the 20,000 most common words as our vocabulary for Cornell. All out of vocabulary words are replaced by the symbol ⟨UNK⟩. The GRU unit has a 1-layer structure with 512 hidden cells. The parameters of the CNQG model are updated by the Adam Optimizer [8] with gradient clipping. We train all models for at most 20 epochs. The learning rate is set to 0.002 and the mini-batch size is fixed to 64. We refer to the Bahdanau Attention Mechanism [1] for decoding.

# 4 RESULTS AND DISCUSSION

## 4.1 Performance on question generation

To answer RQ1, we investigate the appropriateness and informativeness of the questions generated by CNQG and the baselines

[3] https://www.nltk.org/
[4] https://nlp.stanford.edu/projects/glove/

**Table 1: Dataset statistics including properties of different question patterns and context lengths.**

| Dataset | Properties of question pattern | | | | | | | | Context lengths | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | yes/no | what | why | how | who | where | when | which | Min. | Max. | Avg. |
| DailyDialog | 50.60% | 21.12% | 5.07% | 13.56% | 1.58% | 3.18% | 3.46% | 1.43% | 1 | 155 | 10.56 |
| Cornell | 53.90% | 19.18% | 7.33% | 7.67% | 5.36% | 4.09% | 2.02% | 0.45% | 0 | 317 | 9.30 |

in terms of BLEU-1, BLEU-2, Distinct-1 and Distinct-2. We also use a significance test for the difference between the performance of CNQG and the performance of the best performing baseline in terms of BLEU-1 and BLEU-2. The results are presented in Table 2. In general, CNQG consistently achieves the best performance on both datasets in terms of all metrics, which demonstrates its effectiveness for generating appropriate and informative questions. Particularly, the improvements of CNQG over the best performaing baseline in terms of BLEU-1 and BLEU-2 are statistically significant.

On the DailyDialog dataset, context-sensitive approaches, like DCGM-I, HRED and CNQG, achieve obviously higher Distinct scores than NQG, which indicates that the conversational context benefits generating different words and leads to a more informative question in dialogues. But DCGM-I and HRED achieve ver different results in terms of BLEU scores. Many questions generated by HRED are logically reasonable but quite different from the ground-truth, which may explain its poor performance in terms of BLEU scores. We can observe similar results on Cornell. However, for all discussed models, the performance on Cornell in terms of the Distinct scores are worse than on DailyDialog. This may be attributed to the fact that the sentences in Cornell tend to have more uninformative expressions than in DialyDialog, which makes it harder to generate informative and diverse questions [11].

**Table 2: Performance of different question generation models. The results produced by the best baseline and the best performing model in each column are underlined and boldfaced, respectively; * denotes significantly better than the best baseline in a paired $t$-test ($p \leq 0.01$); "DD" is short for DailyDialog.**

| | Model | BLEU-1 | BLEU-2 | Distinct-1 | Distinct-2 |
|---|---|---|---|---|---|
| **DD** | NQG | 0.1683 | 0.0147 | 0.0009 | 0.0024 |
| | DCGM-I | 0.1724 | 0.0154 | 0.0187 | 0.0518 |
| | HRED | 0.0732 | 0.0097 | 0.0199 | 0.0512 |
| | CNQG | **0.2198*** | **0.0366*** | **0.0357** | **0.1299** |
| **Cornell** | NQG | 0.1500 | 0.0067 | 0.0029 | 0.0053 |
| | DCGM-I | 0.1877 | 0.0171 | 0.0093 | 0.0270 |
| | HRED | 0.0782 | 0.0102 | 0.0153 | 0.0287 |
| | CNQG | **0.2109*** | **0.0269*** | **0.0304** | **0.0769** |

## 4.2 Performance on pattern prediction

To answer RQ2, we zoom in on a comparison between CNQG and the baselines in terms of variety and consistency with the ground-truth of generated question patterns. On the test sets, we calculate the question quantity of each pattern for various models as well as for the ground-truth. The results are plotted in Fig. 2.

As shown in Fig. 2, focusing on the pattern variety of generated questions, we see that CNQG and HRED generate more diverse patterns than DCGM-I and NQG. Especially for infrequent patterns like *when*, *why*, *where*, *who* and *which*, DCGM-I and NQG fail to generate those patterns; they are restricted to a single pattern, for instance, NQG only generates *what* patterns on DailyDialog. As for the consistency with the ground-truth, CNQG covers all almost varieties of patterns that exist in the ground-truth, while HRED generates many *what* patterns, with large gaps on Cornell. On both datasets, NQG and DCGM-I lack many patterns that are present in the ground-truth. In addition, for CNQG, HRED and DCGM-I, we can find some instances of *others* patterns that are not identified as questions. By manual inspection, we also found that most of these instances actually correspond to the *yes/no* pattern, which has the most ambiguous interrogatives. For instance, a generated question like "*you have a company?*" does not have any explicit interrogatives, so it is hard to identify its pattern automatically.

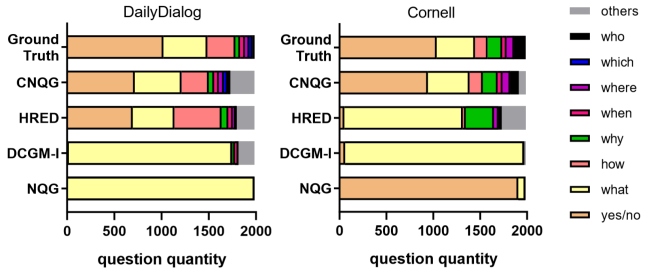



**Figure 2: Comparison on question quantity of each pattern for different models and the ground-truth.**

## 4.3 Impact of context length

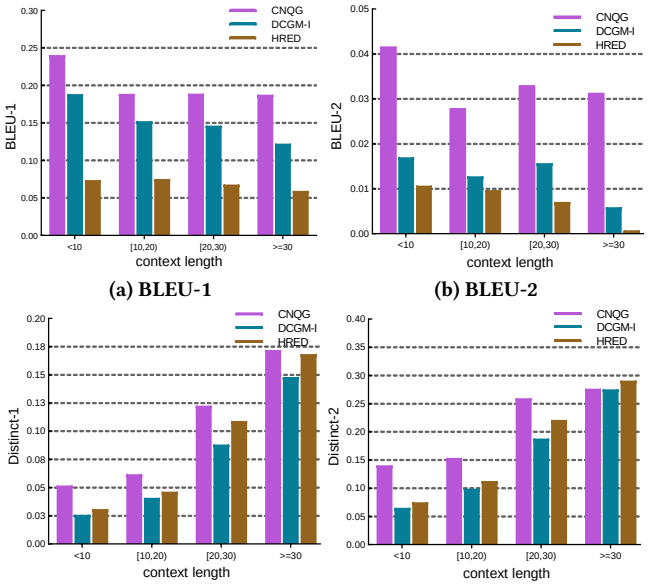
To answer RQ3, we analyze the performance of CNQG and the context-sensitive baselines, i.e., DCGM-I and HRED, on test samples with varying context lengths (measured in number of words). For brevity, we only present our experimental results on the DailyDialog dataset as qualitatively similar phenomena can be found on the Cornell dataset.

We split the test samples into groups according to their context length and present the distribution of tests by context length in Table 3. The majority of the tests are associated with a short context of less than 20 words, which are more likely to be ambiguous. Next, we evaluate the model performance in terms of BLEU-1, BLEU-2, Distinct-1 and Distinct-2, respectively, and plot the results in Fig. 3.

**Table 3: Ratio of test samples with different context length in the testset of DailyDialog.**

| Context length | <10 | [10,20) | [20,30) | ≥30 |
|---|---|---|---|---|
| Ratio | 53.30% | 32.90% | 9.97% | 3.83% |

Generally, for most cases, CNQG outperforms the baselines at every context length in terms of all metrics (except Distinct-2 at length more than 30), which confirms the robustness of CNQG across different context lengths. In particular, for contexts of length less than 10, CNQG clearly outperforms the baselines, more so than for other lengths, demonstrating its effectiveness for short contexts.




(a) BLEU-1 (b) BLEU-2

(c) Distinct-1 (d) Distinct-2

**Figure 3: Comparison on model performance under different context lengths.**

DCGM-I is the best baseline in terms of BLEU scores while HRED is the best in terms of Distinct scores; this is consistent with their overall performance shown in Table 2.

Additionally, with the increase in context length, all models show an increase in terms of the Distinct scores and a decrease in terms of BLEU scores. This indicates that it is increasingly hard for question generation to balance sentence diversity and similarity to the ground-truth when the context length grows, since a long context may introduce various topics while injecting noise at the same time. CNQG uses the semantic information contained in long contexts to achieve high Distinct scores, while it manages to filter out diverging topics so as to maintain a good performance in terms of BLEU scores.

## 4.4 Case study

To obtain a better understanding of the models discussed, we perform a case study by randomly sampling three examples from the datasets we use in our experiments; see Table 4.

In Example 1, the post "*i do not know.*" appears in a dialogue, which is a common but meaningless expression; CNQG, DCGM-I and HRED are able to generate more informative questions than NQG. This could be attributed to the use of the conversational context. As for appropriateness, CNQG performs best among the four models as it can accurately generate a pivotal topic in the question (*job*), which has appeared in the conversational context. In Example 2, all generated questions by the models seem reasonable according to the post and context. However, according to the pattern of the ground-truth question, CNQG obtains a more appropriate question pattern (*why*) than the baseline models. In Example 3, we see that the post only has a single meaningless word while the context provides a useful topic (*soda*). Based on the limited amount of information that is available, the three baseline models fail to generate relevant or informative questions with the correct topic. However, CNQG successfully introduces a highly related topic *beer* and brings out a positive turn for the dialogue; this confirms the effectiveness of CNQG at avoiding breakdown of the dialogue.

**Table 4: Examples of generated questions. Question pattern is bolded and topics underlined.**

| | | |
|---|---|---|
| **Example 1** | Context: | there are different ways of conducting a job search, do you know what you want to do? |
| | Post: | i do not know. |
| | Reference: | **can** you work part-time or full-time? |
| | CNQG: | **did you** have any job? |
| | S2S+Att. | **what**? |
| | DCGM-I | **what** kind of personality do you like? |
| | HRED: | **have you** got any plans? |
| **Example 2** | Context: | look this window is open. |
| | Post: | did you open it before we left? |
| | Reference: | **are you** kidding? it is winter **why** would i open it? |
| | CNQG: | **why** do you think i got to do that? |
| | S2S+Att. | **what about**? |
| | DCGM-I | **what** is the problem? |
| | HRED: | no **why**? |
| **Example 3** | Context: | got any cold soda? |
| | Post: | eh. |
| | Reference: | soda, you got any soda? |
| | CNQG: | you want a beer? |
| | S2S+Att. | you are not locking me. |
| | DCGM-I | **what**? |
| | HRED: | **what**? |

## 5 CONCLUSIONS

To the best of our knowledge, ours is the first work on question generation in the setting of multi-turn open-domain dialogue systems. In this paper, we have proposed a context-augmented neural question generation model CNQG that leverages the conversational context to generate appropriate and informative questions. Experiments on two publicly available conversational datasets provide experimental evidence for the effectiveness of our proposal, showing that CNQG outperforms state-of-the-art question generation baselines in terms of BLEU-1, BLEU-2, Distinct-1 and Distinct-2. CNQG is able to extract useful features from long conversational contexts while maintaining robust performance on short contexts.

As to future work, we want to exploit knowledge bases to enrich interactions in a question-based manner, while maintaining semantic coherence [23]. Also, for dialogues in an e-commerce context we aim to enrich question generation with contrastive questions so as to increase diversity, especially for short contexts [9, 16].

## ACKNOWLEDGMENTS

We thank the anonymous reviewers for their constructive comments. This research was supported by the National Natural Science Foundation of China under No. 61702526, the Defense Industrial Technology Development Program under No. JCKY2017204B064, Ahold Delhaize, the Association of Universities in the Netherlands (VSNU), and the Innovation Center for Artificial Intelligence (ICAI). All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

## REFERENCES


[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *ICLR 2015*.

[2] Kenneth Ward Church and Patrick Hanks. 1989. Word Association Norms, Mutual Information and Lexicography. In *ACL 1989*. 76–83.

[3] Cristian Danescu-Niculescu-Mizil and Lillian Lee. 2011. Chameleons in Imagined Conversations: A New Approach to Understanding Coordination of Linguistic Style in Dialogs. In *CMCL@ACL 2011*. 76–87.

[4] Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to Ask: Neural Question Generation for Reading Comprehension. In *ACL 2017*. 1342–1352.

[5] Wenpeng Hu, Bing Liu, Jinwen Ma, Dongyan Zhao, and Rui Yan. 2018. Aspect-based Question Generation. In *ICLR 2018*.

[6] Shaojie Jiang, Pengjie Ren, Christof Monz, and Maarten de Rijke. 2019. Improving Neural Response Diversity with Frequency-Aware Cross-Entropy Loss. In *WWW 2019*. 2879–2885.

[7] Pei Ke, Jian Guan, Minlie Huang, and Xiaoyan Zhu. 2018. Generating Informative Responses with Controlled Sentence Function. In *ACL 2018*. 1499–1508.

[8] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR 2015*.

[9] Wenqiang Lei, Xiangnan He, Yisong Miao, Qingyun Wu, Richang Hong, Min-Yen Kan, and Tat-Seng Chua. 2020. Estimation-Action-Reflection: Towards Deep Interaction Between Conversational and Recommender Systems. In *WSDM 2020*. 304–312.

[10] Wenqiang Lei, Xisen Jin, Min-Yen Kan, Zhaochun Ren, Xiangnan He, and Dawei Yin. 2018. Sequicity: Simplifying Task-oriented Dialogue Systems with Single Sequence-to-Sequence Architectures. In *ACL 2018*. 1437–1447.

[11] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A Diversity-Promoting Objective Function for Neural Conversation Models. In *NAACL HLT 2016*. 110–119.

[12] Jiwei Li, Alexander H. Miller, Sumit Chopra, Marc'Aurelio Ranzato, and Jason Weston. 2017. Learning through Dialogue Interactions by Asking Questions. In *ICLR 2017*.

[13] Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. DailyDialog: A Manually Labelled Multi-turn Dialogue Dataset. In *IJCNLP 2017*. 986–995.

[14] Nasrin Mostafazadeh, Ishan Misra, Jacob Devlin, Margaret Mitchell, Xiaodong He, and Lucy Vanderwende. 2016. Generating Natural Questions About an Image. In *ACL 2016*. 1802–1813.

[15] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *ACL 2002*. 311–318.

[16] Anna Sepliarskaia, Julia Kiseleva, Filip Radlinski, and Maarten de Rijke. 2018. Preference elicitation as an optimization problem. In *RecSys 2018*. 172–180.

[17] Iulian Vlad Serban, Alberto García-Durán, Çaglar Gülçehre, Sungjin Ahn, Sarath Chandar, Aaron C. Courville, and Yoshua Bengio. 2016. Generating Factoid Questions With Recurrent Neural Networks: The 30M Factoid Question-Answer Corpus. In *ACL 2016*.

[18] Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C. Courville, and Joelle Pineau. 2016. Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Models. In *AAAI 2016*. 3776–3784.

[19] Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. A Neural Network Approach to Context-Sensitive Generation of Conversational Responses. In *NAACL HLT 2015*. 196–205.

[20] Xingwu Sun, Jing Liu, Yajuan Lyu, Wei He, Yanjun Ma, and Shi Wang. 2018. Answer-focused and Position-aware Neural Question Generation. In *EMNLP, 2018*. 3930–3939.

[21] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. In *NeurIPS 2014*. 3104–3112.

[22] Duyu Tang, Nan Duan, Tao Qin, and Ming Zhou. 2017. Question Answering and Question Generation as Dual Tasks. *CoRR* abs/1706.02027 (2017).

[23] Svitlana Vakulenko, Maarten de Rijke, Michael Cochez, Vadim Savenkov, and Axel Polleres. 2018. Measuring Semantic Coherence of a Conversation. In *ISWC 2018*. 634–651.

[24] Svitlana Vakulenko, Kate Revoredo, Claudio Di Ciccio, and Maarten de Rijke. 2019. QRFA: A Data-Driven Model of Information-Seeking Dialogues. In *ECIR 2019*. 541–557.

[25] Wenjie Wang, Minlie Huang, Xin-Shun Xu, Fumin Shen, and Liqiang Nie. 2018. Chat More: Deepening and Widening the Chatting Topic via A Deep Model. In *SIGIR 2018*. 255–264.

[26] Yansen Wang, Chenyi Liu, Minlie Huang, and Liqiang Nie. 2018. Learning to Ask Questions in Open-domain Conversational Systems with Typed Decoders. In *ACL 2018*. 2193–2203.

[27] Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou, and Wei-Ying Ma. 2017. Topic Aware Neural Response Generation. In *AAAI 2017*. 3351–3357.