Low-Resource Generation of Multi-hop Reasoning Questions

Jianxing Yu, Wei Liu, Shuang Qiu, Qinliang Su[*]**, Kai Wang, Xiaojun Quan, Jian Yin**
School of Data and Computer Science, Sun Yat-sen University
Guangdong Key Laboratory of Big Data Analysis and Processing, China
{yujx26, liuw259, qiush9, suqliang}@mail.sysu.edu.cn
{wangk73@mail2, quanxj3@mail, issjyin@mail}.sysu.edu.cn

Abstract

This paper focuses on generating multi-hop reasoning questions from the raw text in a low resource circumstance. Such questions have to be syntactically valid and need to logically correlate with the answers by deducing over multiple relations on several sentences in the text. Specifically, we first build a multi-hop generation model and guide it to satisfy the logical rationality by the reasoning chain extracted from a given text. Since the labeled data is limited and insufficient for training, we propose to learn the model with the help of a large scale of unlabeled data that is much easier to obtain. Such data contains rich expressive forms of the questions with structural patterns on syntax and semantics. These patterns can be estimated by the neural hidden semi-Markov model using latent variables. With latent patterns as a prior, we can regularize the generation model and produce the optimal results. Experimental results on the HotpotQA data set demonstrate the effectiveness of our model. Moreover, we apply the generated results to the task of machine reading comprehension and achieve significant performance improvements.

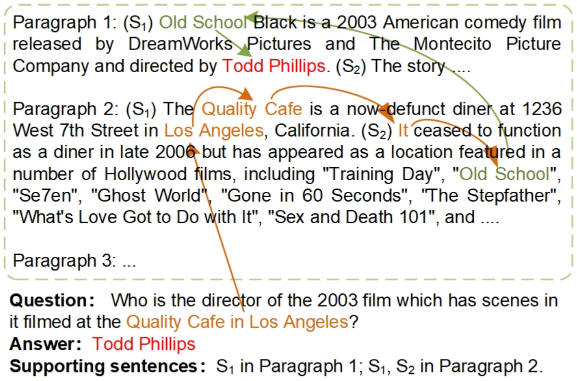
1 Introduction

Question generation (QG) is a hot research topic that aims to create valid and fluent questions corresponding to the answers by fully understanding the semantics on a given text. QG is widely used in many practical scenarios: including providing practice exercises from course materials for educational purposes (Lindberg et al., 2013), initiating the dialog system by asking questions (Mostafazadeh et al., 2017), and reducing the labor cost of creating large-scale labeled samples for the QA task (Duan et al., 2017). The mainstream QG methods can be summarized into the rule-based and neural-based models. The first method often transforms the input text into an intermediate symbolic representation, such as a parsing tree, and then convert the resulting form into a question by well-designed templates or general rules (Hussein et al., 2014). Since rules and templates are hand-crafted, the scalability and generalization of this method are limited. Respectively, the neural model usually directly maps the text to question based on neural network (Du and Cardie, 2017), which is entirely data-driven with far less labor. Such a model can be typically regarded as learning a one-to-one mapping between the text and question. The mapping is mainly used to generate simple questions with a single sentence. However, due to the lack of fine-grained modeling on the evidential relations on the text, such a method has minimal capability to form the multi-hop questions that require sophisticated reasoning skills. These questions have to be grammatically valid. Besides, they need to logically correlate with the answers by deducing over multiple entities and relations in several sentences and paragraphs of the given text. As shown in Fig.(1), the question asks the director of a film, where the film was shot at the Quality Cafe in Los Angeles and Todd Phillips directed it. These two relations can form a reasoning chain from question to answer by logically integrating the pieces of evidence "*Los Angeles*," "*Quality Cafe*," and "*Old School*" as well as the pronoun "*it*" distributed across S_1 in paragraph 1 and S_1, S_2 in paragraph 2. Without capturing such a chain, it is difficult to precisely produce the multi-hop question by using "*Old School*" as a bridging evidence and marginal entity "*Todd Phillips*" as the answer.

For the task of multi-hop QG, a straightforward solution is to extract a reasoning chain from the input text. Under the guidance of the reasoning chain, we learn a neural QG model to make the result satisfy the logical correspondence with the answer. However, the neural model is data-hungry, and the scale of training data mostly limits its performance.

[*] Corresponding author.

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 6729–6739
July 5 - 10, 2020. ©2020 Association for Computational Linguistics



Paragraph 1: (S_1) Old School Black is a 2003 American comedy film released by DreamWorks Pictures and The Montecito Picture Company and directed by Todd Phillips. (S_2) The story ....

Paragraph 2: (S_1) The Quality Cafe is a now-defunct diner at 1236 West 7th Street in Los Angeles, California. (S_2) It ceased to function as a diner in late 2006 but has appeared as a location featured in a number of Hollywood films, including "Training Day", "Old School", "Se7en", "Ghost World", "Gone in 60 Seconds", "The Stepfather", "What's Love Got to Do with It", "Sex and Death 101", and ....

Paragraph 3: ...

Question: Who is the director of the 2003 film which has scenes in it filmed at the Quality Cafe in Los Angeles?
Answer: Todd Phillips
Supporting sentences: S_1 in Paragraph 1; S_1, S_2 in Paragraph 2.

Figure 1: Sample that requires reasoning skills.

Each training example is a triple combined with the text, answer, and question. Since labeling such a combination is labor-intensive, it is difficult to ensure that we can always obtain sufficient training data in real-world applications. We thus formalize the problem as the low-resource generation of multi-hop questions, which is less explored by existing work. This task has substantial research value since reasoning is crucial in quantifying the high-level cognitive ability of machines, and low resource is the key to promote the extensive application. In order to address the problem, we propose to utilize unlabeled data, which is usually abundant and much easier to obtain. Although such data does not combine the questions with the texts and answers, the unlabeled questions contain plentiful expressive forms with structural patterns on the syntax and semantics. These patterns can be seen as the "template" to produce the questions. Thus, we can use the patterns as the prior to regularize the QG model and obtain better results accordingly.

Motivated by the above observations, we propose a practical two-stage approach to learn a multi-hop QG model from both a small-scale labeled data and a large-size unlabeled corpus. In particular, we first exploit the neural hidden semi-Markov model (Dai et al., 2016) to parameterize the sophisticated structural patterns on the questions by latent variables. Without domain knowledge, the variables can be estimated by maximizing the likelihood of the unlabeled data. We then heuristically extract a reasoning chain from the given text and build a holistic QG model to generate a multi-hop question. The evidential relations in the reasoning chain are leveraged to guide the QG model, so as to let the generated result meet multi-hop logical correspondence with the answer. Simultaneously, we naturally incorporate the prior patterns into the QG model. In this way, we can regularize the model and inform it to express a question reasonably. That can improve the syntactic and semantic correctness of the result. With the parameterized patterns, the whole model can be learned from the labeled and unlabeled data in an end-to-end and explainable manner. In order to better balance the supervision of the labeled data and the usage of prior patterns, we propose to optimize the model by reinforcement learning with an augmented evaluated loss. Experiments are conducted on the HotpotQA (Yang et al., 2018) data set, which contains a large number of reasoning samples with manual annotation. Evaluated results in terms of automatic metrics and human judgment show the effectiveness of our approach. Moreover, we apply our generated results to the task of machine reading comprehension. We view the results as pseudo-labeled samples to enrich the training data for the task. That can alleviate the labeled data shortage problem and boost the performance accordingly. Extensive experiments are performed to show the efficacy of our approach in this application with the help of low-resource QG.

The main contributions of this paper include,

- We dedicate to the topic of low-resource generation of multi-hop questions from the text.
- We propose a practical approach to generate multi-hop questions with a minimal amount of labeled data. The logical rationality of the results is guided by the reasoning chain extracted from the text. Besides, the results are regularized to ensure the correctness of syntax and semantics by using the prior patterns estimated from a large-size of unlabeled data.
- We show the potential of our approach in a real-world application on machine reading comprehension by using the generated results.

The rest of this paper is organized as follows. Section 2 elaborates on the proposed low-resource QG framework. Section 3 presents experimental results, while Section 4 shows the QG application and demonstrates its usefulness. Section 5 reviews related works and Section 6 concludes this paper.

2 Approach

In this section, we first define some notations and then present the details of the proposed QG framework, including the learning of prior patterns from the unlabeled data, and the multi-hop QG network guided by the reasoning chain and prior patterns.

2.1 Notations and Problem Formulation

Let $D_L = \{(B_i, A_i, Y_i)\}_{i=1}^{n}$ denote a small set of labeled data that consists of n examples on the text B, answer A, and question Y. Besides, we assume that there are a large number of unlabeled data $D_U = \{Q_j\}_{j=1}^{\mathbb{N}}$ available, where $Q_j \in D_U$ shares the same characteristics with $Y_i \in D_L$ and $\mathbb{N} > n$. Each text contains multiple paragraphs and sentences, involving several logically correlated entities. We aim to generate the new question Y' and answer A' given the evaluated text B' by a QG model, where the answer A' often is a salient entity in the text B'. The question Y' is produced by finding the best $\hat{Y}$ to maximize the conditional probability in $\hat{Y} = \arg\max_{Y'} \prod_{t=1}^{T} p(y_t|B', A', Y'_{<t})$, where $Y'_{<t}$ represents the outputted 1^{th} to $(t-1)^{th}$ terms, y_t is the t^{th} term. The question has to be syntactically and semantically correct. Also, it needs to correspond to the answer by logically deducing over multiple evidential entities and relations scattered across the text. Since the resource in D_L may not be enough to support accurately learning of the $p(\cdot)$, we transfer the linguistic knowledge in D_U and combine it with D_L to enhance the training.

2.2 Learning Patterns from Unlabeled Data

The expressive pattern on the question can be viewed as a sequence of groups. Each group contains a set of term segments that are semantically and functionally similar. Such segmentation is not explicitly given but can be inferred from the text's semantics. It is difficult to characterize this structural pattern by simple hand-crafted rules, while we do not have extra labeled data to learn the pattern by the methods like Variational Auto-Encoder (VAE) (Kingma and Welling, 2014). In order to tackle this problem, we propose to employ the neural hidden semi-Markov model. The model parameterizes the similar segments on the input questions by probabilistic latent variables. Through unsupervised learning, these variables can be trained on the unlabeled data. That can well represent the intricate structural patterns without domain knowledge. Besides, the variables can be incorporated into the generation model naturally, which makes the results more interpretable and controllable.

Given a question Q with a sequence of terms $\{q_t\}_{t=1}^{T}$, we model its segmentation by two variables, including a deterministic state variable $z_t \in \{1, \cdots, K\}$ that indicates the segment to which the t^{th} term belongs, and a length variable $l_t \in \{1, \cdots, L\}$, which specifies the length of the current segment. We assume the question is generated based on a joint distribution as Eq.(1) by multi-step emissions, where $\mathrm{i}(\cdot)$ is the index function; the index on t^{th} term is $\mathrm{i}(t) = \sum_{j=1}^{t} l_j$, with $\mathrm{i}(0) = 0$ and $\mathrm{i}(T') = T$; $q_{\mathrm{i}(t-1)+1:\mathrm{i}(t)}$ is the sequence of terms $(q_{\mathrm{i}(t-1)+1}, \cdots, q_{\mathrm{i}(t)})$. That is, we first produce a segment based on the latent state z_t, and then emits term with a length of l_t on that segment.

$$\prod_{t=0}^{T'-1} p(z_{t+1}, l_{t+1}|z_t, l_t) \prod_{t=1}^{T'} p(q_{\mathrm{i}(t-1)+1:\mathrm{i}(t)}|z_t, l_t) \tag{1}$$

$p(z_{t+1}, l_{t+1}|z_t, l_t)$ is the transition distribution, where the $(t+1)^{th}$ latent state and length are conditioned on their previous ones. Since the length mainly depends on the segment, we can further factorize the distribution as $p(l_{t+1}|z_{t+1}) \times p(z_{t+1}|z_t)$. $p(l_{t+1}|z_{t+1})$ is the length distribution, and we fix it to be uniform up to a maximum length L. In this way, the model can be encouraged to bring together the functionally similar emissions of different lengths. $p(z_{t+1}|z_t)$ is the state distribution, which can be viewed as a $K \times K$ matrix, where each row sums to 1. We define this matrix to be Eq.(2), where e_o, e_j, $e_k \in \mathbb{R}^d$ are the embeddings of the state o, j, k respectively, and $b_{o,j}$, $b_{o,k}$ are the scalar bias terms. Since the adjacent states play different syntactic or semantic roles in the expressive patterns, we set $b_{o,o}$ as negative infinity to disable self-transition. We apply a row-wise *softmax* to the resulting matrix to obtain the desired probabilities.

$$p(z_{t+1} = j|z_t = o) = \frac{\exp(e_j^{\mathsf{T}} e_o + b_{o,j})}{\sum_{k=1}^{K} \exp(e_k^{\mathsf{T}} e_o + b_{o,k})} \tag{2}$$

$p(q_{\mathrm{i}(t-1)+1:\mathrm{i}(t)}|z_t, l_t)$ is the term emission distribution conditioned on a latent state and a length. Based on the Markov process, the distribution can be written as a product over the probabilities of all the question terms, as Eq.(3). In order to compute the term probability, we leverage a neural decoder like the Gated Recurrent Unit (GRU) (Cho et al., 2014). We first formulate the hidden vector h_t^j for yielding j^{th} term as $h_t^j = GRU(h_t^{j-1}, [e_{z_t}; e_{q_{\mathrm{i}(t-1)+j-1}}])$, where $[\cdot;\cdot]$ is a concatenation operator, $e_{q_{\mathrm{i}(t-1)+j-1}}$ and e_{z_t} are the embedding of the term and corresponding segment, respectively. By attending over the given question using h_t^j, we can produce a context vector v_t^j, as $g_{z_t} \odot h_t^j$, where $\odot$ refers to the element-wise multiplication, g_{z_t} is a gate for the latent state z_t, and there are K gate vectors as trainable parameters. We then pass the vector v_t^j through a *softmax* layer to obtain the desired distribution as $p(q_{\mathrm{i}(t-1)+j}|q_{\mathrm{i}(t-1)+j-1}, z_t, l_t) = softmax(\mathrm{W}_q v_t^j + b_q)$, where W_q and b_q are the trainable parameters.

$$\begin{aligned} p(q_{\mathrm{i}(t-1)+1:\mathrm{i}(t)}|z_t, l_t) &= p(q_{\mathrm{i}(t-1)+1}|z_t, l_t) \\ \times \textstyle\prod_{j=2}^{l_t} & p(q_{\mathrm{i}(t-1)+j}|q_{\mathrm{i}(t-1)+j-1}, z_t, l_t) \end{aligned} \qquad (3)$$

Considering that the latent variables are unobserved, we then learn the model by marginalizing over these variables to maximize the log marginal-likelihood of the observed question sequence Q, i.e., $max(logp(Q))$. $p(Q)$ can be formulated as Eq.(4) by the backward algorithm (Murphy, 2002), with the base cases $\beta_T(o) = 1, \forall o \in \{1, \cdots, K\}$. The quantities in Eq.(4) are obtained from a dynamic program, which is differentiable. Thus, we can estimate the model's parameters from the unlabeled data D_U by back-propagation.

$$\begin{aligned} \beta_t(o) &= p(q_{t+1:T}|z_t = o) \\ &= \textstyle\sum_{k=1}^{K} \beta_t^*(k) p(z_{t+1} = k|z_t = o) \\ \beta_t^*(k) &= p(q_{t+1:T}|z_{t+1} = k) \\ &= \textstyle\sum_{j=1}^{L} [\beta_{t+j}(k) p(l_{t+1} = j|z_{t+1} = k) \\ &\qquad p(q_{t+1:t+j}|z_{t+1} = k, l_{t+1} = j)] \\ p(Q) &= \textstyle\sum_{k=1}^{K} \beta_0^*(k) p(z_1 = k) \end{aligned} \qquad (4)$$

2.3 Multi-hop QG Net with Regularization

Afterward, we incorporate the learned patterns into the generation model as the prior. Such prior can be acted as a soft template to regularize the model. That can ensure the correctness of the results in syntax and semantics, especially when the labeled data is insufficient to learn the correspondence between the text and question. Fig.(2) illustrates the architecture of our model. We first estimate the prior pattern by sampling a sequence of latent states z with the length l. We then extract the reasoning chain and other textual contents involved in asking and solving a specific question from the given text. Under the guidance of both the reasoning chain and the prior patterns, we build a multi-hop QG model on the extracted contents by the sequence-to-sequence framework (Bahdanau et al., 2015). The evidential relations in the chain are used to enhance the logical rationality of the results. The prior pattern helps to facilitate the performance in low-resource conditions by specifying the segmentation of the generated results.

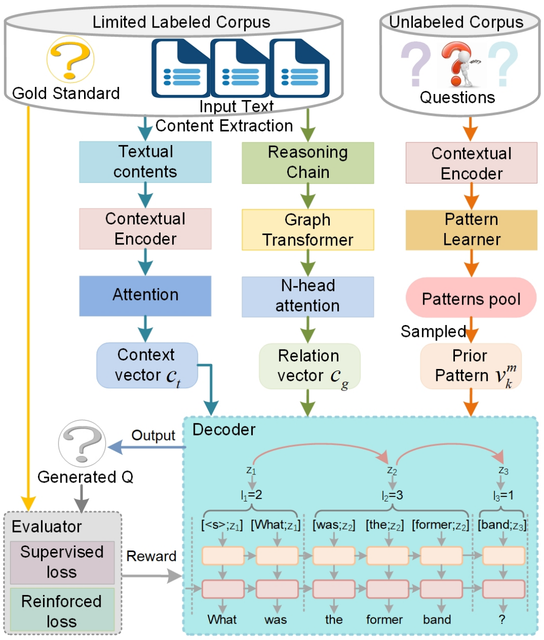


Figure 2: Flow chart of the low-resource multi-hop QG network.

2.3.1 Prior Patterns Estimation

Using the Viterbi algorithm (Zucchini et al., 2016), we can obtain the typed segmentation of a given question. Such segmentation can be characterized by a sequence of latent states z. Each segment, like the phrase, is associated with a state, reflecting that the state frequently produces that segment. Based on the labeled data D_L, we can collect all sequences of latent states, which can be seen as a pool of prior patterns. We sample one from the pool uniformly. And then, we view it as a question template with S distinct segments, as $\{< z_t^k, l_t^k >\}_{k=1}^{S}$, where z_t is a state variable for the t^{th} term, l_t is the length variable derived by the probability $p(l_t|z_t)$, z_t^k and l_t^k are obtained by collapsing adjacent z_t and l_k with the same value. In order to easily incorporate into the generation model, we encode the template as a vector $v_k^m = g_{z_t} \odot h_k^m$, where h_k^m is the hidden vector for generating m^{th} term, as $GRU(h_{k-1}^m, [e_{z_m}; e_{y_{t-1}}])$, m satisfies $\mathrm{i}(m-1) < t \leq \mathrm{i}(m), k = t - \mathrm{i}(m-1)$.

2.3.2 Question-Related Content Extraction

Given a text, we use the method proposed by Yu et al. (2020) to extract the question-related content. In order to make the paper self-contained, we briefly describe the approach in this section. It first extracted the entities from the text, and view them as the potential answers and evidences. It then links the entities to create a graph by three kinds of relations, including dependency, coreference, and synonym. Based on the graph, it heuristically extracts a sub-graph as the reasoning chain. The textual contents on the sub-graph are then gathered, including the answer, reasoning type, evidential entities, and sentences on the entities. The extraction is based on three question types, consisting of the *Sequence*, *Intersection*, and *Comparison*. These types account for a large proportion of the multi-hop questions on most typical data sets, for example, 92% in HotpotQA data set (Min et al., 2019).

2.3.3 Question Generation with Guidance

We then develop a multi-hop QG model based on the extracted contents. This model is guided by the reasoning chain and prior pattern, so that the generated results are not only logical but also fluent. In the pre-processing phase, we first mask the answer from the input contents by a special token <UNK>, to avoid the answer inclusion problem (Sun et al., 2018). That is, the answer words may appear in the question that would reduce the rationality.

Encoder: The reasoning chain is encoded via an N head graph transformer (Vaswani et al., 2017), so as to integrate all evidential relations fully. Each node is represented by contextualizing on its neighbors, as $h_v^g = e_v + \|_{n=1}^N \sum_{j\in\mathcal{N}_v} a^n(e_v, e_j)\mathrm{W}^n e_j$, where $\|$ denotes the concatenation, e_v is the embedding of node's entity, $a^n(\cdot,\cdot)$ is n^{th} head attention function, $\mathcal{N}_v$ is the set of neighbors. By aggregation with N-head attention, we can get a relation-aware vector c_g as Eq.(5), where $\mathrm{W}_g^n, \mathrm{W}_h, \mathrm{W}_d$ are trainable matrices, $\mathcal{C}$ is the set of nodes in the chain.

$$\begin{aligned}
a^n(s_t, h_v^g) &= \frac{exp((\mathrm{W}_h h_v^g)^\mathsf{T}\mathrm{W}_d s_t)}{\sum_{k\in\mathcal{N}_v} exp((\mathrm{W}_h h_k^g)^\mathsf{T}\mathrm{W}_d s_t)} \\
c_g &= s_t + \|_{n=1}^N \sum_{v\in\mathcal{C}} a^n(s_t, h_v^g)\mathrm{W}_g^n h_v^g
\end{aligned} \tag{5}$$

Other textual inputs are encoded in two steps: (1) each text term is embedded by looking up the pre-trained vectors, such as BERT (Devlin et al., 2019). (2) The resulting embeddings are fed into a bi-directional GRU to incorporate a sequential context. In detail, the sentences are represented by concatenating the final hidden states of GRU, as $[\overleftarrow{h_1^b}; \overrightarrow{h_J^b}]$, where j^{th} term is $h_j^b = [\overleftarrow{h_j^b}; \overrightarrow{h_j^b}]$, $\overleftarrow{h_j^b} = GRU(e_j^b, \overleftarrow{h_{j+1}^b})$, $\overrightarrow{h_j^b} = GRU(e_j^b, \overrightarrow{h_{j-1}^b})$; $[\cdot;\cdot]$ denotes the concatenation of two vectors; e_j^b is the augmented embedding of j^{th} term; J is the size of all terms. Similarly, the answer and evidence entities are integrally encoded as $h^a = [\overleftarrow{h_1^a}; \overrightarrow{h_O^a}]$.

Attention: For the textual inputs, we fully integrate the encodings and their correlations by attention. First, we use self-attention (Wang et al., 2017) to grasp the long-term dependency in the sentences, as $[\hat{h}_j^b]_{j=1}^J = SelfAttn([h_j^b]_{j=1}^J)$. Subsequently, we exploit multi-perspective fusion (Song et al., 2018) to grasp the answer-related context in the sentences and strengthen their cross interactions. That is, $[h_j^{b'}]_{j=1}^J = MulPerFuse([\hat{h}_j^b]_{j=1}^J, [h_o^a]_{o=1}^O)$. By aggregating the significant information over all the terms, we can obtain a context vector c_t as Eq.(6), where α_{tj} is the normalized attention weight, a_{tj} denotes the alignment score, s_t refers to the t^{th} hidden state of the decoder, $v, b, \mathrm{W}_s, \mathrm{W}_b$ are trainable parameters.

$$\begin{aligned}
a_{tk} &= v^\mathsf{T} tanh(\mathrm{W}_s s_t + \mathrm{W}_b h_k^{b'} + b) \\
\alpha_{tj} &= \exp(a_{tj}) / \textstyle\sum_{k=1}^J \exp(a_{tk}) \\
c_t &= \textstyle\sum_{j=1}^J \alpha_{tj} h_j^{b'}
\end{aligned} \tag{6}$$

Decoder: Based on the context vector c_t, we exploit another GRU as the decoder. Each question term is yielded by the distribution in Eq.(7), where ρ is a 1-dim embedding of the reasoning type, W_o and b_o are trainable parameters. We use a copy mechanism (Gu et al., 2016) to tackle unknown words problem, where $p_{copy}(\cdot)$ denotes the copy distribution. In order to let the questions logically correlate with answers, we guide the decoder by the vector c_g, which encodes the reasoning chain. Accordingly, we regularize the model to adaptively fit the prior pattern represented by the vector v_k^m. That can improve the generated quality when the labeled data is insufficient.

$$\begin{aligned}
&p_{voc}(y_t) = Softmax(\mathrm{W}_o[s_t; c_t; c_g; \rho] + b_o) \\
&p_{copy} = \textstyle\sum_{j=1}^J \alpha_{tj} \times \mathbb{1}\{y == w_j\} \\
&p_g = Sigmoid(c_t, s_t, y_{t-1}) \\
&s_t = GRU(s_{t-1}, v_k^m) \\
&p(y_t) = p_g \cdot p_{voc}(y_t) + (1 - p_g) \cdot p_{copy}(y_t)
\end{aligned} \tag{7}$$

2.3.4 Learning with Limited Labeled Data

A straightforward solution to train the above QG model is the supervised learning. It minimizes the cross-entropy loss at each generated term by referring to the ground-truth in the labeled data D_L, as $\mathcal{L}_{sl} = -\frac{1}{n}\sum_{i\in D_L}\sum_{t=1}^{T_i} \log p(y_{it}|Y_{i;<t}, A_i, B_i)$. However, since D_L only contains a few samples, we would not have enough supervision from D_L to get the best results. While we leverage the unlabeled data D_U to facilitate the training, it is difficult to subtly balance the supervised signal from

D_L and the prior pattern learned from D_U. In order to address the problem, we resort to reinforcement learning. It can globally measure the overall quality of the results by minimizing the loss $\mathcal{L}_{rl} = -\mathbb{E}_{Y^s \sim \pi_\theta}[r(Y^s)]$, where Y^s is a sampled result, Y^* is the ground-truth, θ is the parameters of the QG model, and π is the generation policy of the model. $r(\cdot)$ is a function to evaluate the generated quality. It is the weighted sum of three rewards, including (a) *Fluency*: we calculate the negative perplexity (Zhang and Lapata, 2017) of Y^s by a BERT-based language model p_{LM}, that is, $-2^{-\frac{1}{T}\sum_{t=1}^{T}\log_2 p_{LM}(y_t|Y^s{}_{<t})}$; (b) *Answerability*: we use a metric $QBLEU_4(Y^s, Y^*)$ (Nema and Khapra, 2018) to measure the matching degree of Y^s and Y^* by weighting on several answer-related factors, including question type, content words, function words, and named entities; (c) *Semantics*: we employ word movers distance (WMD) (Gong et al., 2019) to measure the predicted result Y^s, which has different expressive forms but same semantics with gold Y^*, as $-WMD(Y^s, Y^*)/Length(Y^*)$, where $Length(\cdot)$ is the length function used as the normalization factor. By considering the metrics are non-differentiable, we exploit the policy gradient method (Li et al., 2017) for optimization. In order to enhance readability, we train the model by a mixed loss, as $\mathcal{L} = \gamma\mathcal{L}_{rl} + (1-\gamma)\mathcal{L}_{sl}$, where γ is a trade-off factor.

3 Evaluations

We extensively evaluate the effectiveness of our approach, including the comparisons with state-of-the-art and the application on a task of MRC-QA.

3.1 Data and Experimental Settings

The evaluations were performed on three typical data sets, including HotpotQA (Yang et al., 2018), ComplexWebQuestions (Talmor and Berant, 2018), and DROP (Dua et al., 2019). These data sets were collected by crowd-sourcing, consisting of 97k, 35k, and 97k examples, respectively. The HotpotQA data set contained a large proportion of labeled examples. Each comprised of the question, answer, and text with several sentences. Therefore, the HotpotQA data set was suitable to evaluate the multi-hop QG task. The other two data sets contained abundant reasoning questions, but they are not associated with the text and answer. We thus viewed them as the unlabeled data. In order to simulate the low-resource setting, we randomly sampled 10% of the HotpotQA train set to learn the models, and evaluated them on the test set with a size of 7k. We verified the generated quality for each evaluated method by comparing the matching degree between the result and gold-standard. We adopted three standard evaluation metrics in the QG task, including *BLEU-4* (Papineni et al., 2002), *METEOR* (Banerjee and Lavie, 2005), and *ROUGE-L* (Lin, 2004). Furthermore, we carried out human evaluations to analyze the generated results. To avoid biases, we randomly sampled 100 cases from the test set and generated questions for each test case by all the evaluated methods. We then invited eight students to give the binary rating on each question independently. The rating was in terms of three metrics, including valid *syntax*, *relevance* to input textual sentences, and logical *rationality* to the answer. We averaged the cumulative scores of the 100 binary judgments as the performances corresponding to the evaluated methods. The resultant scores were between 0∼100, where 0 is the worst, and 100 is the best. We used Randolph's free-marginal kappa (Randolph, 2005) to measure the agreements among the raters.

Model configurations were set as follows. We leveraged 768-dimension pre-trained vectors from the uncased BERT to embed words. The number of states K and emissions L in the semi-Markov model was set to 50, 4, respectively. The size of hidden units in both encoder and decoder was 300. The recurrent weights were initialized by a uniform distribution between -0.01 and 0.01 and updated with stochastic gradient descent. We used Adam (Kingma and Ba, 2015) as the optimizer with a learning rate of 10^{-3}. The trade-off parameter γ was set to 0.4. For pattern learning, we parsed every question by the Stanford CoreNLP toolkit (Manning et al., 2014). We then learn better segmentation by forcing the model not to break syntactic elements like the VP and NP. To reduce the bias, we carried out five runs and reported the average performance.

3.2 Comparisons on QG State-of-the-Arts

We compared our approach against five typical and open-source methods. These methods were based on the sequence-to-sequence framework with attention. According to the different techniques used, we summarized them as follows. (a) the basic model with the copy mechanism, i.e., **NQG++** (Zhou et al., 2017); (b) **ASs2s** (Kim et al., 2019), which encoded the answer separately to form answer-focused questions; (c) **CorefNQG** (Du and Cardie, 2018) that incorporated linguistic features to represent the inputs better; (d) **MaxPointer** (Zhao et al., 2018) using gated self-attention to form questions for long text inputs; (e) **MPQG+R** (Song et al., 2018) that captured a broader context in the text to produce the context-dependent results. In order to understand the effect of unlabeled data, we examined two variants of the proposed model. That is, **Ours-Pattn** which was trained without unlabeled data, and **Ours-50%** that used 50% unlabeled data for training. Moreover, we performed empirical ablation studies to gain better insight into the relative contributions of various components in our model, including **Ours-Chain** that discarded the guidance of the reasoning chain vector and **Ours-Reinf** that replaced the reinforcement learning with a simple supervised learning.

Table 1: Comparisons of our approach against baselines. Statistically significant with t-test, p-value<0.01.

Methods	BLEU-4	METEOR	ROUGE-L
NQG++	14.55	15.01	31.85
ASs2s	16.89	17.04	34.92
CorefNQG	16.16	16.53	34.30
MaxPointer	17.08	17.34	35.38
MPQG+R	14.90	15.46	32.39
Ours	**19.07**	**19.16**	**39.41**
Ours-50%	18.33	18.36	37.85
Ours-Pattn	17.10	17.35	35.40
Ours-Chain	18.11	18.18	37.37
Ours-Reinf	18.22	18.39	37.48

As reported in Tab.(1), our approach achieved the best performance. We significantly outperformed the best baseline (i.e., **MaxPointer**) by over 11.6%, 10.5%, 11.4% in terms of *BLEU-4*, *METEOR*, and *ROUGE-L*, respectively. From the comparisons among **Ours-Pattn**, **Ours-50%**, and **Ours**, we found that the performance improves with more unlabeled data. Although we lack an appropriate comparative model based on the unlabeled data, these results can still indicate the effectiveness of our model. With only limited labeled data, our model can effectively leverage unlabeled data to guide the generation. Besides, the ablation on all evaluated components led to a significant performance drop. We may infer that the reasoning chain is crucial for multi-hop QG on the guidance of logical correlations. Also, the reinforcement learning can globally optimize the model by balancing the prior patterns and labeled supervision.

3.3 Human Evaluations and Analysis

Tab.(2) illustrated the results of human evaluation. The average kappa were all above 0.6, which indicated substantial agreement among the raters. Consistent with quantitatively analyzed results in Section 3.2, our model significantly outperformed all baselines in terms of three metrics, where the improvement on the *rationality* metric was the largest. That showed the satisfied quality of our generated results, especially in terms of multi-hop ability.

Table 2: Human evaluations and kappa agreement. Ration. is short for the rationality metric. Statistically significant with t-test, p-value<0.01.

Methods	Syntax	Relevance	Ration.	Kappa
NQG++	54.3	44.8	50.3	0.61
ASs2s	61.3	50.8	55.8	0.62
CorefNQG	59.0	49.4	54.8	0.64
MaxPointer	61.8	51.8	56.5	0.63
MPQG+R	55.5	47.5	51.3	0.64
Ours	**68.3**	**57.3**	**62.3**	0.65

3.4 Evaluations on Value of Unlabeled Data

We investigated the value of unlabeled data for the overall performance, especially when the labeled data was inadequate. In particular, we randomly sampled $\{10\%, 40\%, 70\%, 100\%\}$ of the labeled data, and split the unlabeled data into ten subsets. For each scale on the labeled data, we incrementally added by one subset of unlabeled data to learn the QG model. We used the same training protocol and reported the overall performance on the test set. As shown in Fig.(3), even a small amount of unlabeled data can play a decisive role in improving the performance in terms of three metrics. The ratio of improvement was higher when the scale of the labeled data was small. The results further verified the usefulness of unlabeled data on learning the QG model with a low labeled resource.

3.5 Evaluations on the Mixed Loss Objective

In order to examine the gains of our training approach with the mixed loss objective, we tuned the trade-off parameter (i.e., γ) from $[0, 1]$ with 0.1 as an interval. The performance change curve was displayed in Fig.(4). The best performance was obtained at $\gamma = 0.4$. The performance dropped dramatically when γ was close to 0 or 1. We would infer that both objectives could help to measure the quality of the outputted results better, and thus train the model efficiently.

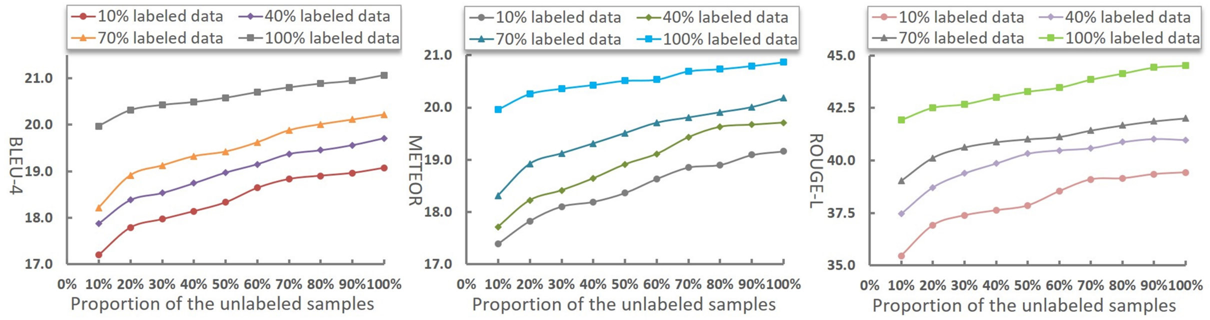


Figure 3: Evaluations on effectiveness of unlabeled data under different scales of labeled data.

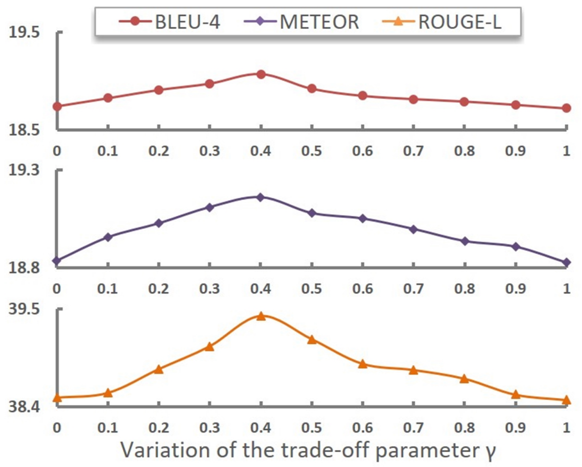


Figure 4: Evaluations on mixed objective trade-off.

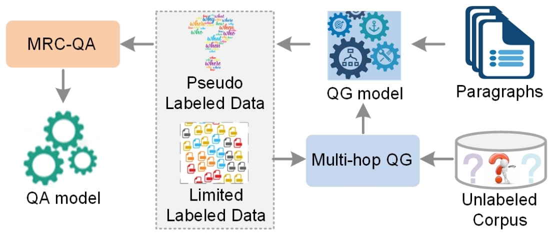


Figure 5: Apply multi-hop QG to support MRC-QA.

4 Application on the Task of MRC-QA

The task of machine reading comprehension (MRC-QA) aims to answer given questions by understanding the semantics of the text. The mainstream methods are based on the neural network. These methods often need a lot of labeled data for training, but the data is expensive to obtain. Thus, we are inspired to apply our generated results to enrich the training set for the task of MRC-QA. Fig.(5) demonstrates the architecture of this application. Given a case from a small-size labeled set, we first extracted the contents correlated to a specific question from the case's text, including the reasoning chain, reasoning type, answer, evidential entities, and sentences on the entities. We then learned our QG model based on the contents and generated questions as pseudo data to augment the labeled set. For each evaluated case, we could yield approximately 5∼8 pseudo samples consisted of the text, question, and answer. Later, we trained an MRC-QA model on the augmented labeled set and reported the performance on the test set. By referring to the leaderboard on the HotpotQA website, we chose an open-source model for MRC-QA, named Dynamically Fused Graph Network (DFGN) (Qiu et al., 2019), which achieved the state-of-the-art at the paper submission time. Considering that the size of the training set impacted the model's performance, we ran the entire pipeline with different proportions of the labeled data, so as to verify the proposed model thoroughly. Two evaluation metrics were employed, including exact match (*EM*) and *F1*. We examined the tasks of answer span extraction, supporting sentence prediction, and the joint task in the distractor setting.

Table 3: Comparison of our QG+QA model against the QA model under different proportions of labeled data.

Labeled	Answer span		Support pred.		Joint	
Data#	EM	F1	EM	F1	EM	F1
QG + QA(i.e., the DFGN model)						
10%	.501	.633	.469	.764	.277	.509
20%	.551	.672	.500	.801	.317	.574
30%	.567	.697	.520	.815	.339	.600
40%	.569	.704	.521	.829	.340	.615
50%	.571	.713	.531	.833	.344	.619
60%	.586	.717	.533	.834	.346	.624
70%	.593	.729	.535	.839	.353	.626
80%	.606	.731	.540	.845	.356	.630
90%	.610	.741	.550	.853	.359	.632
100%	.614	.746	.558	.858	.360	.635
QA(i.e., the DFGN model)						
100%	.563	.697	.515	.816	.336	.598

Tab.(3) showed that our QG+QA model trained on 30% labeled data obtained competitive performance against the QA model learned on the 100% labeled data. When using more labeled data, the performance advantages of our QG+QA model continued to grow. Such results showed that our QG model could enlarge the coverage and diversity of the MRC-QA training set given limited labeled data. That could help to learn the state-of-the-art. Moreover, we conducted case studies to understand the generating behavior vividly. As exhibited in Tab.(4), our QG model could generate massive questions on multi-hop reasoning. Contrastively, the gold standard often contained one sample since it was labor-intensive to enumerate all the cases.

Table 4: Case studies on our multi-hop QG model.

Passage: *... (S_1) 'The Hard Easy' is the episode written by Thomas Herpich. (S_2) He was born in October, 1979 in Torrington, Connecticut, American, along with his twin brother Peter who was a painter and artist. (S_3) Thomas is best known for being a storyboard artist on the animated television series 'Adventure Time'. ...*
Results of Ours Method
Question: *When was the birth time for the writer of the episode 'The Hard Easy'?* **Answer**: *October, 1979*
Question: *Where is the birthplace for the writer of the episode 'The Hard Easy'?* **Answer**: *Torrington, Connecticut*
Question: *What nationality was the writer of the episode 'The Hard Easy'?* **Answer**: *American*
Question: *Who is the twin brother for the writer of the episode 'The Hard Easy'?* **Answer**: *Peter*
Question: *What is the occupation for the twin brother of the episode writer of 'The Hard Easy'?* **Answer**: *painter and artist*
Gold Standard
Question: *Who is the brother for the writer of the episode 'The Hard Easy'?* **Answer**: *Peter*

5 Related Works

Existing models for the QG task include rule-based and neural-based methods. Since the rules are handcrafted, the first method is of low scalability (Chali and Hasan, 2015). The researcher turns to the neural model. It can directly map the inputs into questions by using an attention-based sequence-to-sequence framework, which is entirely data-driven with far less labor. Various techniques have been applied to this framework, including answer separately encoding, using linguistic features, capturing border context, reinforcement learning, and emphasizing on question-worthy contents (Pan et al., 2019). These methods are mainly used to generate simple questions with a single sentence (Yu et al., 2019). They are challenging to generate the reasoning questions accurately due to the lack of fine-grained modeling on the evidential relations in the text. In order to address the problem, Yu et al. (2020) proposed to incorporate a reasoning chain into the sequential framework, so as to guide the generation finely. All the methods are built of the assumption that sufficient labeled data is available. However, labeled data is quite scarce in many real-world applications (Yang et al., 2019). The low-resource problem has been studied in the tasks such as machine translation (Gu et al., 2018), pos tagging (Kann et al., 2018), word embedding (Jiang et al., 2018), text generation (Wiseman et al., 2018), and dialogue systems (Mi et al., 2019). To the best of our knowledge, the low-resource multi-hop QG is untouched by existing work. We thus focus on this topic and propose a method to fulfill the gap.

6 Conclusions and Future Works

We have proposed an approach to generate the questions required multi-hop reasoning in low-resource conditions. We first built a multi-hop QG model and guided it to satisfy the logical rationality by the reasoning chain extracted from a given text. In order to tackle the labeled data shortage problem, we learned the structural patterns from the unlabeled data by the hidden semi-Markov model. With the patterns as a prior, we transferred this fundamental knowledge into the generation model to produce the optimal results. Experimental results on the HotpotQA data set demonstrated the effectiveness of our approach. Moreover, we explored the generated results to facilitate the real-world application of machine reading comprehension. We will investigate the robustness and scalability of the model.

Acknowledgments

This work is supported by the Key-Area Research and Development Program of Guangdong Province (2018B010107005, 2019B010120001), National Natural Science Foundation of China (61906217,61806223,61902439,U1711262,U161 1264,U1711261,U1811261,U1811264,U1911203), National Key R&D Program of China (2018YFB1 004404), Guangdong Basic & Applied Research Foundation (2019B1515130001), Fundamental Re search Funds for Central Universities (19lgpy219).

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of ICLR*, pages 124–131, San Diego, CA, USA.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan.

Yllias Chali and Sadid A. Hasan. 2015. Towards topic-to-question generation. *Computational Linguistics*, 41(1):1–20.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Proceedings of the 2014 EMNLP*, pages 1724–1734, Doha, Qatar.

Hanjun Dai, Bo Dai, Yan-Ming Zhang, Shuang Li, and Le Song. 2016. Recurrent hidden semi-markov model. In *Proceedings of the ICLR*, pages 14–23, San Juan, Puerto Rico.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 NAACL*, pages 4171–4186, Minneapolis, Minnesota.

Xinya Du and Claire Cardie. 2017. Identifying where to focus in reading comprehension for neural question generation. In *Proceedings of the 2017 EMNLP*, pages 2067–2073, Copenhagen, Denmark.

Xinya Du and Claire Cardie. 2018. Harvesting paragraph-level question-answer pairs from Wikipedia. In *Proceedings of the 56th ACL*, pages 1907–1917, Melbourne, Australia.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 NAACL*, pages 2368–2378, Minneapolis, Minnesota.

Nan Duan, Duyu Tang, Peng Chen, and Ming Zhou. 2017. Question generation for question answering. In *Proceedings of the 2017 EMNLP*, pages 866–874, Copenhagen, Denmark.

Hongyu Gong, Suma Bhat, Lingfei Wu, JinJun Xiong, and Wen-mei Hwu. 2019. Reinforcement learning based text style transfer without parallel training corpus. In *Proceedings of the 2019 NAACL*, pages 3168–3180, Minneapolis, Minnesota.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th ACL*, pages 1631–1640, Berlin, Germany.

Jiatao Gu, Yong Wang, Yun Chen, Victor O. K. Li, and Kyunghyun Cho. 2018. Meta-learning for low-resource neural machine translation. In *Proceedings of the 2018 EMNLP*, pages 3622–3631, Brussels, Belgium.

Hafedh Hussein, Mohammed Elmogy, and Shawkat Guirguis. 2014. Automatic english question generation system based on template driven scheme. In *International Journal of Computer Science Issues*, pages 45–53.

Chao Jiang, Hsiang-Fu Yu, Cho-Jui Hsieh, and Kai-Wei Chang. 2018. Learning word embeddings for low-resource languages by PU learning. In *Proceedings of the 2018 NAACL*, pages 1024–1034, New Orleans, Louisiana.

Katharina Kann, Johannes Bjerva, Isabelle Augenstein, Barbara Plank, and Anders Søgaard. 2018. Character-level supervision for low-resource POS tagging. In *Proceedings of the Workshop on Deep Learning Approaches for Low-Resource NLP*, pages 1–11, Melbourne.

Yanghoon Kim, Hwanhee Lee, Joongbo Shin, and Kyomin Jung. 2019. Improving neural question generation using answer separation. In *Proceedings of the Thirty-Third AAAI*, pages 6602–6609, New York City, NY, USA.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of ICLR*, pages 324–331, San Diego, CA, USA.

Diederik P Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *Proceedings of the ICLR*, pages 224–231, Banff, AB, Canada.

Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017. Adversarial learning for neural dialogue generation. In *Proceedings of the 2017 EMNLP*, pages 2157–2169, Copenhagen, Denmark.

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

David Lindberg, Fred Popowich, John Nesbit, and Phil Winne. 2013. Generating natural language questions to support learning on-line. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 105–114, Sofia, Bulgaria.

Christoper Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of the 52nd ACL*, pages 55–60, Baltimore, Maryland.

Fei Mi, Minlie Huang, Jiyong Zhang, and Boi Faltings. 2019. Meta-learning for low-resource natural language generation in task-oriented dialogue systems. In *Proceedings of the IJCAI*, pages 3151–3157, Macao, China.

Sewon Min, Victor Zhong, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2019. Multi-hop reading comprehension through question decomposition and rescoring. In *Proceedings of the 57th ACL*, pages 6097–6109, Florence, Italy.

Nasrin Mostafazadeh, Chris Brockett, Bill Dolan, Michel Galley, Jianfeng Gao, Georgios Spithourakis, and Lucy Vanderwende. 2017. Image-grounded conversations: Multimodal context for natural question and response generation. In *Proceedings of the Eighth IJCNLP*, pages 462–472, Taipei, Taiwan.

Kevin Murphy. 2002. Hidden semi-markov models (hsmms). In *unpublished notes*.

Preksha Nema and Mitesh M. Khapra. 2018. Towards a better metric for evaluating question generation systems. In *Proceedings of the 2018 EMNLP*, pages 3950–3959, Brussels, Belgium.

Liangming Pan, Wenqiang Lei, Tat-Seng Chua, and Min-Yen Kan. 2019. Recent advances in neural question generation. In *arXiv preprint arXiv:1905.08949*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th ACL*, pages 311–318, Philadelphia, Pennsylvania, USA.

Lin Qiu, Yunxuan Xiao, Yanru Qu, Hao Zhou, Lei Li, Weinan Zhang, and Yong Yu. 2019. Dynamically fused graph network for multi-hop reasoning. In *Proceedings of the 57th ACL*, pages 6140–6150, Florence, Italy.

Justus J Randolph. 2005. Free-marginal multirater kappa (multirater kfree): An alternative to fleiss' fixed-marginal multirater kappa.

Linfeng Song, Zhiguo Wang, Wael Hamza, Yue Zhang, and Daniel Gildea. 2018. Leveraging context information for natural question generation. In *Proceedings of the 2018 NAACL*, pages 569–574, New Orleans, Louisiana.

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

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. In *Proceedings of the 2018 NAACL*, pages 641–651, New Orleans, Louisiana.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st NIPS*, pages 6000–6010, Vancouver, BC, Canada.

Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. 2017. Gated self-matching networks for reading comprehension and question answering. In *Proceedings of the 55th ACL*, pages 189–198, Vancouver, Canada.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2018. Learning neural templates for text generation. In *Proceedings of the 2018 EMNLP*, pages 3174–3187, Brussels, Belgium.

Ze Yang, Wei Wu, Jian Yang, Can Xu, and Zhoujun Li. 2019. Low-resource response generation with template prior. In *Proceedings of the 2019 EMNLP)*, pages 1886–1897, Hong Kong, China.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 EMNLP*, pages 2369–2380, Brussels, Belgium.

Jianxing Yu, Quan Xiaojun, Su Qinliang, and Yin Jian. 2020. Generating multi-hop reasoning questions to improve machine reading comprehension. In *Proceedings of the WWW*, pages 550–561, Taipei, Taiwan.

Jianxing Yu, Zheng-Jun Zha, and Tat-Seng Chua. 2012. Answering opinion questions on products by exploiting hierarchical organization of consumer reviews. In *Proceedings of the EMNLP*, pages 391–401, Jeju Island, Korea.

Jianxing Yu, Zhengjun Zha, and Jian Yin. 2019. Inferential machine comprehension: Answering questions by recursively deducing the evidence chain from text. In *Proceedings of the 57th ACL*, pages 2241–2251, Florence, Italy.

Xingxing Zhang and Mirella Lapata. 2017. Sentence simplification with deep reinforcement learning. In *Proceedings of the 2017 EMNLP*, pages 584–594, Copenhagen, Denmark.

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

Qingyu Zhou, Nan Yang, Furu Wei, Chuanqi Tan, Hangbo Bao, and Ming Zhou. 2017. Neural question generation from text: A preliminary study. In *Proceedings of NLPCC*, pages 662–671, Dalian, China.

Walter Zucchini, Iain L MacDonald, and Roland Langrock. 2016. Hidden markov models for time series: an introduction using r. In *Chapman and Hall/CRC*.