Latent Reasoning for Low-Resource Question Generation

Xinting Huang,[1] Jianzhong Qi,[1] Yu Sun,[2] Rui Zhang[3]*
[1]The University of Melbourne, [2]Twitter Inc., [3]www.ruizhang.info
{xintingh@student., jianzhong.qi@}unimelb.edu.au,
ysun@twitter.com, rui.zhang@ieee.org

Abstract

Multi-hop question generation requires complex reasoning and coherent language realization. Learning a generation model for the problem requires extensive multi-hop question answering (QA) data, which are limited due to the manual collection effort. A two-phase strategy addresses the insufficiency of multi-hop QA data by first generating and then composing single-hop sub-questions. Learning this generating and then composing two-phase model, however, requires manually labeled question decomposition data, which is labor intensive. To overcome this limitation, we propose a novel generative approach that optimizes the two-phase model without question decomposition data. We treat the unobserved sub-questions as latent variables and propose an objective that estimates the true sub-questions via variational inference. We further generalize the generative modeling to single-hop QA data. We hypothesize that each single-hop question is a sub-question of an unobserved multi-hop question, and propose an objective that generates single-hop questions by decomposing latent multi-hop questions. We show that the two objectives can be unified and both optimize the two-phase generation model. Experiments show that the proposed approach outperforms competitive baselines on HOTPOTQA, a benchmark multi-hop question answering dataset.

1 Introduction

Question generation aims to automatically generate valid and coherent questions based on given context, which is widely applied to enrich question answering (QA) datasets, facilitate text comprehension (Ko et al., 2020), seek clarification in conversation (Rao and Daumé III, 2019), etc. Recently, neural encoder-decoder based approaches have shown promising results for simple, single-hop question generation (Du and Cardie, 2018). Such approaches directly maps context (e.g., text passages) to questions without *reasoning*, and thus struggle when generating multi-hop questions (Pan et al., 2020). Here, reasoning refers to identifying and aggregating the relevant information taken from multiple documents to derive the question. Table 1 illustrates the reasoning process of a multi-hop question; in this example, the entity that links the two passages, i.e., "AMG", is firstly identified, and the relations around it in the context are transformed into a question. To model such reasoning processes in an end-to-end manner requires extensive training data, and is thus impractical due to the

Table 1: Multi-Hop Question Reasoning Example

Supportive Evidence	**Paragraph A. Dario Franchitti**
	[1] George ..., known professionally as Dario Franchitti, is a retired Scottish racing driver. [2] After Franchitti did not secure a single-seater drive in 1995, he was *contracted* by the AMG team to *compete* in touring cars in the DTM and its successor — the International Touring Car Championship.
	Paragraph B. Mercedes-AMG
	[1] Mercedes-AMG GmbH (AMG)... is the high performance division of Mercedes-Benz. [2] Mercedes-AMG is *headquartered* in Affalterbach, Baden Württemberg, Germany.
Reasoning Progress	< Dario Franchitti , contracted by , AMG > < Dario Franchitti , competed in , DTM > < AMG , headquartered in , Affalterbach, Baden Württemberg, Germany >
Multi-hop Question	After he was contracted by the team that is headquartered in Affalterbach, Baden Württemberg, Germany, Dario Franchitti competed in what series?
Sub-Questions	Which team is headquartered in Affalterbach, Baden Württemberg, Germany? After contracted by AMG, Dario Franchitti competed in what series?
Answer	DTM

*Rui Zhang is the corresponding author.

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 3008–3022

extensive collection effort of multi-hop QA data.

To address this problem, recent studies propose to augment the generation model with an explicit reasoning progress. For example, a straightforward solution is to identify the anchoring entities via named entity recognition (NER), and find relations via relation extraction. The extracted structural reasoning path, in the form of subject-predicate-object triples as illustrated in Table 1, is then fed to the generation model as auxiliary features (Yu et al., 2020b). However, the reasoning capability is constrained by the off-the-shelf extraction tools which cannot be extended to arbitrary context (Yang et al., 2018; Dhingra et al., 2020).

Another line of recent studies on multi-hop question answering models the reasoning process by decomposing a multi-hop question into several sub-questions (Min et al., 2019; Wolfson et al., 2020). As illustrated in Table 1, the answer of the multi-hop question can be derived by answering a series of single-hop sub-questions. Ideally, question generation can also adopt this two-phase strategy which first generates sub-questions and then composes the sub-questions into a multi-hop question. However, this strategy requires a parallel corpus that annotates each multi-hop question to its corresponding sub-questions, and obtaining such annotations still requires extensive efforts and costs.

To address these issues, we propose to jointly optimize the two-phase model using *non-parallel* single-hop and multi-hop corpuses only, in which the questions are not paired. We propose a generative objective that models the multi-hop and single-hop question generation (QG) tasks in a unified way. The key idea is that each question, either multi-hop or single-hop, can be considered a partially observed ⟨multi-hop question, sub-question⟩ pair and treat the unobserved part as a latent variable. In the generative modeling of multi-hop QG, we use the two-phase model as a generation model and introduce a posterior model to estimate unobserved sub-questions. The generation and the posterior models are jointly optimized via variational inference (Kingma and Welling, 2014). For generative single-hop QG, we instead use the two-phase model as a posterior model to estimate unobserved multi-hop questions, and the posterior model is jointly optimized with a generation model that decomposes a multi-hop question into sub-questions. In this way, we integrate the optimization of the two-phase model in both generative multi-hop and single-hop QG tasks, serving as the generation and the posterior model, respectively.

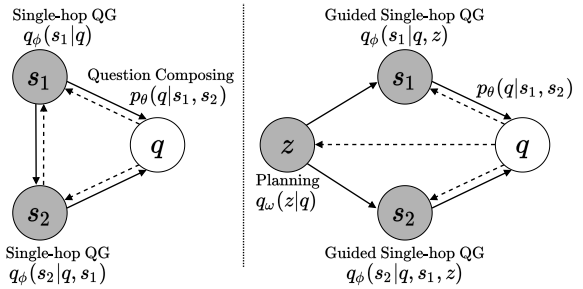
Optimizing the generative objective in the text space is, however, prone to compounding errors due to the diversities of potential reasoning paths. There are multiple ways to raise a single-hop question given the same piece of information, and it is challenging to find the valid one only given the text passages. We address this challenge by equipping the generative modeling with a planning mechanism that uses a latent variable to encode the desired reasoning path. In this way, the inference of sub-questions is guided by a pre-sampled plan (i.e., the latent variable) and thus maintains consistency with the target multi-hop question. We achieve latent variable learning by incorporating an end-to-end differentiable bottleneck into the sub-question generation model, which can be naturally integrated into the overall objective. Moreover, the proposed planning mechanism also promotes a more stable training. This is because the original generative modeling involves a sequential sampling of latent variables (i.e., sub-questions), which is known to cause high variance and result in an unstable training (He et al., 2020). The planning mechanism relieves the sequential sampling requirement, since it encodes the high-level planning and covers the dependency between sub-questions.

Our contributions are summarized as follows:

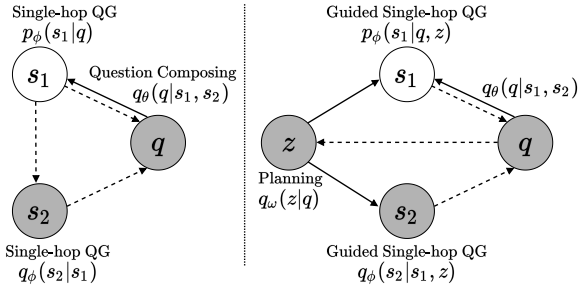
- We propose a novel generative objective that unifies non-parallel question corpuses and relieves the requirements of extensive annotations for learning a two-phase question generation model.
- We propose a planning mechanism to guide the generation towards sub-questions that are more probable to compose into a multi-hop question.
- We conduct experiments on a benchmark multi-hop question answering dataset. The results show that our approach outperforms the state-of-the-art under both language generation and question answering based evaluations.

2 Preliminaries

Let $D_{\mathcal{M}} = \{(q_i, a_i, c_i)|1 \leq i \leq N\}$ be a set of N multi-hop question-answer-evidence triples, where the evidence is a set of potentially relevant sentences $c_i = \{d_1, d_2, ..., d_k\}$, and each multi-hop question q requires reasoning over multiple sentences to find the answer a. Multi-hop question generation (QG) aims to generate a question q that has the pre-selected answer a given the evidence set



(a) Generative multi-hop QG (left) with planning (right)



(b) Generative single-hop QG (left) with planning (right)

Figure 1: A graphical representation of the proposed generative model. White circles denote the observed variables and gray circles denote the latents. [1]

c. Existing studies adopt a strategy commonly used in single-hop question generation, which formulates multi-hop QG as a seq-to-seq problem. Since extensive annotation efforts are needed to produce multi-hop QG examples, few multi-hop QG examples are available. Thus, a naive adoption of seq-to-seq learning may not yield an effective multi-hop QG model, especially in the low-resource scenario.

To address data insufficiency, a two-phase strategy is considered based on the assumption that each multi-hop question q can be decomposed into two single-hop sub-questions s_1 and s_2.[2] The multi-hop question generation is then performed by a sub-question generation model $p_{\mathcal{S}}$ and a question composing model $p_{\mathcal{C}}$ as

$$p(q|a, c) = p_{\mathcal{C}}(q|s_1, s_2)p_{\mathcal{S}}(s_2, s_1|a, c). \quad (1)$$

The training of these two models require question decomposition data, which are pairs of a multi-hop question and its corresponding sub-question annotations $\{(q, \langle s_1, s_2\rangle)\}$. However, it is non-trivial to obtain the question decomposition data, which requires extensive human annotation effort.

[1] Note that the generation of single-hop questions s, multi-hop question q, and planning variables z are conditioned on evidence set c and answer a, which is omitted in Fig. 1.

[2] The formulation can be easily extended to more sub-questions

3 Proposed Model

We take a two-phase approach for multi-hop question generation while do not require a question decomposing dataset that contains pairs of multi-hop questions and sub-questions. We assume that a single-hop question answering dataset $D_{\mathcal{S}}$ and a multi-hop dataset $D_{\mathcal{M}}$ are available for training. Both datasets are non-parallel, i.e., contain question-answer pairs but not sub-questions, and the evidence passages of both datasets shall come from the same source (e.g., Wikipedia articles).

Under these problem settings, we aim to learn the single-hop QG model $p_{\mathcal{S}}$ and the question composing model $p_{\mathcal{C}}$ using both $D_{\mathcal{S}}$ and $D_{\mathcal{M}}$. To effectively train these two models in the absence of question decomposition data, we propose a unified generative formulation that naturally connects single-hop and multi-hop questions. Specifically, in modeling the generation process of multi-hop questions, we treat the corresponding sub-questions as latent variables and propose an objective that jointly optimizes $p_{\mathcal{S}}$ and $p_{\mathcal{C}}$ (Sec. 3.1). We further extend the generative formulation to model the generation of single-hop questions, and both generation processes together form the overall optimization objective (Sec. 3.2). Then, we propose a planning-aware generation strategy to better optimize the objective in Sec. 3.3. We summarize the overall learning and inference process in Sec. 3.4.

3.1 Generative Modeling of Multi-Hop QG

We now reconsider the two-phase question generation strategy in Eqn. 1. Since we do not have the parallel data, it is infeasible to directly model the conditional probability $p(q|s)$, where $s = \{s_1, s_2\}$ is the set of sub-questions of q. We thus propose to treat the unobserved sub-questions as latent variables, and describe $p(q|a, c)$ in a generative way as

$$p(q) = \sum_s p(q, s) = \sum_s p_\theta(q|s)p_\psi(s) \quad (2)$$

where $p(q)$ and $p(q, s)$ are shorthands for $p(q|a, c)$ and $p(q, s|a, c)$, [3] p_ψ is a conditional prior model, and p_θ is a generation model for multi-hop questions. Since this likelihood is intractable, we instead derive and optimize its evidence lower bound

[3] Same below for brevity when the context is clear.

(ELBO) (Kingma and Welling, 2014)

$$\begin{aligned}\log p(q) &\geq \mathbb{E}_{q_\phi(s|q)}[\log \frac{p_\theta(q|s)p_\psi(s)}{q_\phi(s|q)}] \\ &= \mathbb{E}_{q_\phi(s|q)}[\log p_\theta(q|s)] - \mathrm{KL}(q_\phi(s|q)||p_\psi(s))\end{aligned} \tag{3}$$

where $q_\phi(s|q)$ is a posterior model for latent variable s, and KL denotes the Kullback-Leibler divergence. We now substitute the latent variable s with two sub-questions, s_1 and s_2, and define the factorized form of the posterior and the prior in a hierarchical manner

$$\begin{aligned}q_\phi(s|q) &= q_\phi(s_2|q, s_1)q_\phi(s_1|q) \\ p_\psi(s) &= p_\psi(s_2|s_1)p_\psi(s_1).\end{aligned} \tag{4}$$

We can now rewrite the ELBO in Eqn. 3 with the factorization and obtain

$$\begin{aligned}\log p(q) \geq\ & \mathbb{E}_{q_\phi(s_1|q)q_\phi(s_2|s_1,q)}[\log p_\theta(q|s_1, s_2)] \\ & - \mathrm{KL}(q_\phi(s_1|q)||p_\psi(s_1)) \\ & - \mathrm{KL}(q_\phi(s_2|q, s_1)||p_\psi(s_2|s_1)) \\ & \coloneqq \mathcal{L}_{\mathrm{ELBO}}(q)\end{aligned} \tag{5}$$

Fig. 1(a) shows the directed graphical model of the generative modeling of multi-hop question generation. Specifically, given an evidence set and a pre-selected answer, a single-hop question s_1 is first sampled. Given s_1 and relevant information in the context, a second sub-question s_2 that satisfies a valid reasoning process is further sampled. Since two sub-questions are both unobserved, we estimate s_1 and s_2 using the posterior model q_ϕ. The sub-questions then form the observed multi-hop question q via question composing as $p_\theta(q|s_1, s_2)$.

To perform effective optimization, we tie the parameters of the posterior model q_ϕ at different hierarchies, i.e., $q_\phi(s_1|\cdot)$ and $q_\phi(s_2|\cdot)$, as one single-hop QG model. Such parameter tying also applies to the prior model p_ψ. We implement the generation model p_θ, the prior p_ψ, and the posterior q_ϕ in Eqn. 5 using pre-trained encoder-decoder models which will be detailed in Sec. 3.4. We notice that the prior p_ψ and the generation model p_θ actually play the same role as the single-hop QG model $p_\mathcal{S}$ and question composing model $p_\mathcal{C}$ in Eqn. 1. Thus, the generative modeling enables a joint optimization of $p_\mathcal{S}$ and $p_\mathcal{C}$ using multi-hop QA data only and without question decomposing data.

3.2 Generative Modeling of Single-Hop QG

Considering that the multi-hop QA data is limited, we propose to integrate single-hop QA data into the joint optimization objective. We extend the proposed generative modeling by assuming that each single-hop question is obtained by decomposing an unobserved multi-hop question. With a slight abuse of notation, we use (s, a, c) to denote a single-hop question-answer-evidence triple, and describe $p(s|a, c)$ as

$$p(s) = \sum_q p(s, q) = \sum_q p_{\theta'}(s|q)p_{\psi'}(q) \tag{6}$$

where we omit the condition as in Eqn. 2, and q is a multi-hop question that has a sub-question s. The generation model $p_{\theta'}$ and the prior model $p_{\psi'}$ are parameterized with θ' and ψ', respectively. We treat the unobserved q as a latent variable and derive the evidence lower bound as

$$\begin{aligned}\log p(s) \geq\ & \mathbb{E}_{q_{\phi'}(q|s)}[\log p_{\theta'}(s|q)] \\ & -\mathrm{KL}(q_{\phi'}(q|s)||p_{\psi'}(q))] \coloneqq \mathcal{L}_{\mathrm{ELBO}}(s)\end{aligned} \tag{7}$$

where $q_{\phi'}$ is a posterior model to estimate the unobserved question q.

Fig. 1(b) illustrates the generative modeling for single-hop QG. Specifically, a multi-hop question is first sampled by the prior $p_{\psi'}$, and we assume that its sub-question set includes the observed single-hop question s. The question s is then generated by decomposing the multi-hop question q via $p_{\theta'}(s|q)$. We estimate the unobserved multi-hop question q using the posterior model $p_{\phi'}$.

We observe that the posterior approximation in single-hop QG (dashed line in Fig. 1(b)-left) is the same as the generative process in multi-hop QG (solid line in Fig. 1(a)-left). Thus, we can realize the posterior model $q_{\phi'}(q|s)$ by reusing the prior p_ψ and the generative model p_θ in Eqn. 5 as

$$q_{\phi'}(q|s) = p_\theta(q|\hat{s}, s)p_\psi(\hat{s}|s) \tag{8}$$

where $\hat{s}$ is the unobserved second sub-question that forms the multi-hop question together with s. Note that we no longer need a hierarchical form since one sub-question is observed.

Further, we observe that the generative process in single-hop QG (solid line in Fig. 1(b)-left) is part of the posterior approximation of multi-hop QG (dashed line in Fig. 1(b)-left). This way, we realize the generation model $p_{\theta'}$ and the prior $p_{\psi'}$ using the models already present in multi-hop QG

$$\begin{aligned}p_{\theta'}(s|q) &= q_\phi(s|q) \\ p_{\psi'}(q) &= p_\theta(q|s_1, s_2)p_\psi(s_2|s_1)p_\psi(s_1)\end{aligned} \tag{9}$$

Table 2: Question Generation Diversification Example.

Supportive Evidence	
After Franchitti did not secure a single-seater drive in 1995, he was contracted by the AMG team to compete in touring cars in the DTM and its successor — the International Touring Car Championship.	
Potential Generated Sub-Questions	
Q: Did Dario Franchitti secure a single seater drive in 1995?	**A:** No
Q: Dario Franchitti was contracted by which team to compete in the DTM?	**A:** AMG
Q: The International Touring Car Championship is the successor of what series?	**A:** DTM
Q: After contracted by AMG, Dario Franchitti competed in what series?	**A:** DTM

where the prior $p_{\psi'}(q)$ is inferred by first estimating and then composing the latent sub-questions s_1 and s_2. Note that p_θ, p_ψ, and q_ϕ are all taken from the generative modeling of multi-hop QG. Thus, the single-hop QG objective (Eqn. 7) optimizes the same set of models as in multi-hop QG objective (Eqn. 3). This way, we seamlessly unify the multi-hop and single-hop QA data for joint optimization.

3.3 Planning Guided Question Generation

There is a challenge under the generative formulation: the diversification of feasible generated questions can impinge the model training. Given the same evidence set and pre-selected answer, there can be multiple ways to raise a questions (Lee et al., 2020). However, not every potential single-hop question is qualified as a sub-question to form the target multi-hop question, as illustrated in Table. 2. To address this challenge, we propose to learn a latent planning variable which serves as a generation planning to guide the generation process.

The latent planning variable aims to capture the high-level reasoning required to answer the multi-hop questions, which is abstracted as a reasoning path in existing studies. In order to model decision making of the reasoning path, we define the latent variable z as a discrete variable. We now incorporate the latent variable into the generative modeling of multi-hop QG

$$\begin{aligned}\log p(q) &= \log \sum_s \int_z p(q,z,s)dz \\ &\geq \mathbb{E}_{q_\omega(z|q)}[p(q|z)] - \mathrm{KL}(q_\omega(z|q)||p_\omega(z)) \\ &:= \mathcal{L}_{\mathrm{ELBO}}(q,z)\end{aligned} \tag{10}$$

where q_ω and p_ω are posterior and prior models, respectively, and the reason of having the same parameters ω will be detailed later. The conditional probability $p(q|z)$ is modeled by letting the terms of $\mathcal{L}_{\mathrm{ELBO}}(q)$ in Eqn. 5 be additionally conditioned on the sampled latent variable z (as illustrated in Fig. 1(a)-right). The generation of sub-questions, both prior p_ψ and posterior q_ϕ, is now aware of the planning as

$$\begin{aligned}q_\phi(s|q,z) &= q_\phi(s_1|q,z)q_\phi(s_2|q,z) \\ p_\psi(s|z) &= p_\psi(s_1|z)p_\psi(s_2|z).\end{aligned} \tag{11}$$

We now no longer need a hierarchical form like Eqn. 4, since the latent planning variable already encodes the information of the other sub-question. Thus, this formulation also alleviates the high variance issue commonly encountered in hierarchical variational training (Vahdat and Kautz, 2020).

We also consider the planning guided mechanism in the generative modeling of single-hop QG

$$\begin{aligned}\log p(s) &= \log \sum_q \int_z p(s,z,q)dz \\ &\geq \mathbb{E}_{q_\omega(z|q)}[p(s|z)] - \mathrm{KL}(q_\omega(z|s)||p_\omega(z)) \\ &:= \mathcal{L}_{\mathrm{ELBO}}(s,z)\end{aligned} \tag{12}$$

where $p(s|z)$ is modeled by letting the prior and the posterior in $\mathcal{L}_{\mathrm{ELBO}}(s)$ be additionally conditioned on z The realizations in Eqn. 8 and Eqn. 9 are now formulated as

$$\begin{aligned}q_{\phi'}(q|s,z) &= p_\theta(q|\hat{s},s)p_\psi(\hat{s}|z) \\ p_{\psi'}(q|z) &= p_\theta(q|s_1,s_2)p_\psi(s_1|z)p(s_2|z)\end{aligned} \tag{13}$$

We implement the latent variable as discretized VAE (van den Oord et al., 2017) by adding a learnable codebook between the encoder and the decoder. The codebook is a set of prototype vectors $e_k, k \in 1,2...K$, each having the same dimensionality as that of the encoder output. The discrete variable is obtained by using a nearest-neighbor lookup to find the vector closest to the encoder output. The corresponding prototype vector is then fed into the decoder as an additional context embedding to which every decoding step could attend. With this discretization bottleneck design, the encoder-decoder model and the codebook can be jointly optimized.

3.4 Learning and Inference

We initialize the generative model p_θ, the prior p_ψ and the posterior q_ϕ using BART (Lewis et al.,

2020), a pre-trained seq-to-seq model. BART uses the standard Transformer based encoder-decoder architecture (Vaswani et al., 2017), and is optimized by reconstructing the intentionally corrupted documents. We adopt an initial fine-tuning step for all three models using question answering data $D_{\mathcal{S}}$ and $D_{\mathcal{M}}$, which adjusts the initialization pre-trained from general texts to better fit the question generation tasks. We then optimize p_θ, p_ψ, and q_ϕ together with the discretization bottleneck q_ω using the generative modeling of both multi-hop and single-hop question answering data

$$\mathcal{L} = \sum_{q \in D_{\mathcal{M}}} \mathcal{L}_{\text{ELBO}}(q, z) + \sum_{s \in D_{\mathcal{S}}} \mathcal{L}_{\text{ELBO}}(s, z) \quad (14)$$

After training the single-hop QG model (i.e., p_ψ), question composing model (i.e., p_θ), and the bottleneck q_ω, inference follows the two-stage strategy. We first infer a latent planning variable given the evidence set and the answer. The sub-questions are generated based on the inferred planning variable and are composed into a multi-hop question.

4 Experiments

To show the effectiveness of the proposed approach, *planning guided latent reasoning* (PLAR), we experiment on two multi-hop question generation settings (Sec. 4.1). We compare against state-of-the-art approaches in both settings (Sec. 4.2). We further consider a question answering based performance measure, and analyze the effectiveness of the proposed generative modeling (Sec. 4.3).

4.1 Settings

We use HotpotQA (Yang et al., 2018), a crowd-sourced multi-hop question answering (QA) dataset in our experiments. It contains over 90K question answering examples, and the evidence set of each question includes relevant paragraphs from Wikipedia. The question-relevant sentences within these paragraphs are further annotated as supporting facts. We follow the original data split of HotpotQA, which includes 90,440 / 6,072 examples for training and evaluation, respectively. We further hold out 6,072 examples from the training data as the validation set. We use SQuAD (Rajpurkar et al., 2016) as the single-hop QA dataset, which has over 100K questions also crowd-sourced based on Wikipedia articles. Following the conventional evaluation metrics, we use n-gram **BLEU** (Papineni et al., 2002), **METEOR** (Banerjee and Lavie, 2005), and **ROUGE-L** (Lin and Hovy, 2002) to evaluate the question generation quality.

We consider two input settings to thoroughly evaluate the multi-hop question generation (QG) performance: *sentence-level* and *paragraph-level*. In the first setting, following the existing multi-hop QG task formulation (Pan et al., 2020; Yu et al., 2020b), we take the question-relevant sentences (i.e., supporting facts) along with the answer as inputs to generate the question. However, human annotated supporting facts are not always available, while identifying two relevant paragraphs is relatively achievable. Thus, we further consider a paragraph-level setting where, besides the answer, we instead use the paragraphs containing supporting facts as part of the input. In both settings, in order to simulate a low-resource scenario, we train PLAR and other baselines using two different subsets of the question answering examples, HotpotQA-10K and HotpotQA-30K, containing 10K and 30K randomly sampled training examples, respectively.

Note that we do not utilize any annotated question decomposition dataset (e.g., QDMR (Wolfson et al., 2020)). This is because it is labour-intensive to obtain the extra question decomposition annotations, which are not present in HotpotQA. Thus, it is not practical to assume such decomposition annotations would be available in different QA tasks. We aim to tackle this challenge by utilizing non-parallel single-hop questions, which is relatively easy to acquire and do not require extra task-specific annotations.

We compare with three baselines that are based on seq-to-seq models and are competitive in single-hop question generation tasks: **ASs2s** (Kim et al., 2019), **Maxout-QG** (Zhao et al., 2018), **BART** (Lewis et al., 2020). We compare with two baselines that consume auxiliary reasoning path features for multi-hop QG: **RC-QG** (Yu et al., 2020b) uses reasoning chains built via named entity recognition and relation extraction; and **SG-DQG** (Pan et al., 2020) adopts semantic role labeling techniques to build semantic graphs. We also compare the full **PLAR** with its two variants: **Pipeline** individually trains a single-hop QG model and a question composing model using synthetic question decomposition data obtained as Perez et al. (2020); and **PLAR w/o plan** uses the generative objectives as PLAR without the planning mechanism.

Table 3: Question Generation Results (Sentence-Level)

Model		HotpotQA-30K				HotpotQA-10K			
		BLEU-1	BLEU-4	METEOR	ROUGE-L	BLEU-1	BLEU-4	METEOR	ROUGE-L
Seq-to-Seq Application	ASs2s	27.12	10.11	13.27	25.69	24.32	9.27	11.93	23.20
	Maxout-QG	28.21	10.26	13.64	25.80	25.47	9.19	11.84	23.51
	BART	30.52	11.15	14.87	27.22	26.10	10.12	12.46	23.63
Reasoning Path Enhanced	RC-QG	30.86	11.36	15.29	28.66	29.31	11.16	14.88	26.89
	SG-DQG	32.93	12.32	16.40	29.81	31.12	12.27	15.25	27.90
Proposed	Pipeline	30.11	11.65	15.20	27.28	28.23	10.02	13.21	25.21
	PLAR w/o plan	35.19	13.48	17.54	31.02	33.68	13.87	16.64	28.87
	PLAR	**37.32**	**14.94**	**18.87**	**32.63**	**35.96**	**15.32**	**17.37**	**29.85**

4.2 Overall Results

Table 3 shows that PLAR consistently outperforms baselines on both subsets in the sentence-level input setting. We can see that PLAR achieves a significant performance gain for all metrics. For example, PLAR (32.63) outperforms SG-DQG (29.81) under ROUGE-L on HotpotQA-30K. Meanwhile, we also find that the generative modeling is essential to the performance gain of PLAR. For examples, PLAR w/o plan (16.64) achieves 25.9% improvements over Pipeline (13.21) under METEOR on HotpotQA-10K. This validates that unifying single-hop and multi-hop QA data can effectively alleviate the data scarcity issue. We further find that Pipeline has a heavier performance decrease (comparing with the baselines) when having fewer data. For example, Pipeline outperforms RC-QG under BLEU-4 on HotpotQA-30K, while it is outperformed by RC-QG on HotpotQA-10K. This is largely because the training of each phase is individual performed which is prone to data insufficiency especially in a more extreme low-resource scenario.

For the paragraph-level input setting, Table 4 shows the results that PLAR consistently outperforms the baselines by a large margin. For example, PLAR (17.79) achieves a gain of more than 33% compared to SD-DQG (13.37) under METEOR on HotpotQA-30K. By comparing PLAR (27.64) with PLAN w/o plan (24.81) and Pipeline (23.04) under ROUGE-L on HotpotQA-10K, we find that the contribution of the planning mechanism is more significant than that of the generative modeling. This is largely because the diversification of potential sub-questions raises greater challenges in the paragraph-level setting. Using the planning variables, PLAR can effectively generate the feasible sub-questions. We also provide qualitative examples in Appendix to show the effectiveness of the planning variables. We also find that the reasoning path augmented baselines are not as competitive as in the sentence-level input setting. For example, RC-QG outperforms all the seq-to-seq based baselines under METEOR in the sentence-level setting, while it only outperforms ASs2s in the paragraph-level setting. The reason is that handcrafted reasoning features cannot generalize well to a larger evidence set. PLAR overcomes this limitation by optimizing reasoning capability taking advantage of both single-hop and multi-hop QA data.

4.3 Discussion

We first study whether the generated questions can boost the question answering performance. We compare the performances of a BERT QA model (Devlin et al., 2019) on both subsets, where the QA model is trained using QA data generated by different QG models. The results in Table 5 show that the learning of multi-hop QA models relies heavily on sufficient supervision, since a significant perform reduction is observed when training on a subset only. PLAR achieves more effective training than the baselines and its variants, especially in the more challenging subset. It achieves the most performance gain (17.3%) over the subset-only training result under F1 on HotpotQA-30K. We also find that the QG results of BART do not improve QA performance while BART performs comparable to other baselines (e.g., SG-DQG) on automatic evaluation metrics. This is aligned with our intuition that the text fluency is insufficient for obtaining multi-hop questions that benefit the QA task. It is essential to incorporate reasoning into the generation process.

We now study the effect of unified generative question generation. To investigate how the generative multi-hop (Eqn. 5) and single-hop objective (Eqn. 7) contribute to the overall question generation training, we compare PLAR with PLAR using multi-hop objective only and PLAR using planning guided multi-hop objective under varying sizes of single-hop question answering data. The results on

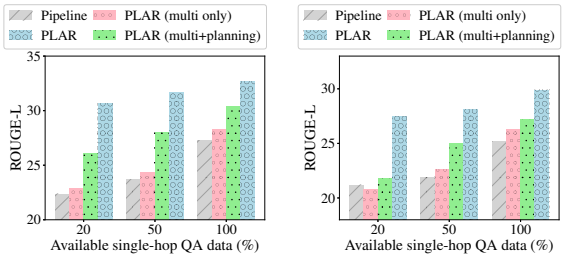
Table 4: Question Generation Results (Paragraph-Level)

Model		HotpotQA-30K BLEU-1	BLEU-4	METEOR	ROUGE-L	HotpotQA-10K BLEU-1	BLEU-4	METEOR	ROUGE-L
Seq-to-Seq Application	ASs2s	23.33	8.71	11.01	22.18	22.01	8.82	10.00	20.76
	Maxout-QG	25.04	9.82	12.36	24.51	23.71	9.05	11.28	22.42
	BART	26.13	10.34	13.40	24.97	24.60	9.63	12.37	22.79
Reasoning Path Enhanced	RC-QG	25.75	9.02	12.51	24.08	23.32	8.71	11.08	20.51
	SG-DQG	26.97	10.42	13.37	25.43	24.07	10.03	11.54	22.57
Proposed	Pipeline	27.06	10.43	13.93	25.62	24.41	10.24	11.84	23.04
	PLAR w/o plan	32.27	12.10	15.76	28.48	27.31	11.23	12.06	24.81
	PLAR	**36.74**	**13.63**	**17.79**	**30.35**	**32.58**	**13.32**	**14.18**	**27.64**

Table 5: Question Answering Results using Synthetic Data from Question Generation

QA Supervision		HotpotQA-30K EM	F1	HotpotQA-10K EM	F1
w/o QG	Subset only	52.2	66.1	46.0	58.9
Subset w/ QG	BART	51.4	64.0	43.1	54.3
	SG-DQG	53.3	67.2	47.3	62.8
	Pipeline	54.8	68.3	49.1	61.7
	PLAR w/o plan	60.9	72.8	53.4	65.0
	PLAR	**61.5**	**73.0**	**54.8**	**66.2**

* Both question generation and question answering are performed in the paragraph-level setting.



(a) Results on HotpotQA-30K (b) Results on HotpotQA-10K

Figure 2: Effects of unified generative modeling

the two subsets under ROUGE-L in the sentence-level setting are shown in Fig. 2(a) and Fig. 2(b). We can see that both objectives are important. For example, when using complete single-hop QA data on HotpotQA-30K, multi-hop and single-hop generative objectives bring 7.3% and 11.5% improvement, respectively. We further find that the performance gain of PLAR is largely attributed to the single-hop generative objective when available single-hop questions are limited. The reason is that without the generative single-hop objective, training the sub-question generation model heavily relies on the initial fine-tuning step, and is thus prone to single-hop QA data insufficiency. The full PLAR model addresses this limitation by further training the sub-question generation model with supervision from generative single-hop and multi-hop QG.

5 Related Work

Question generation has a wide range of applications besides expanding question answering data, such as initiating a conversation of dialogue systems (Mostafazadeh et al., 2017), providing practice exercises for educational purposes (Jia et al., 2020), and accelerating real-time question answering (Seo et al., 2019). It also has great potential in enriching task-oriented dialogue datasets (Sun et al., 2016, 2017; Huang et al., 2020a; Kim et al., 2020b). Early studies build on encoder-decoder models and utilize different evidence information, e.g., Wikipedia passages (Du and Cardie, 2018), reviews (Yu et al., 2020c), and dialogue history (Gao et al., 2019). These studies often assume that the questions are single-hop which be answered by one piece of evidence. As more high-quality multi-hop question answering datasets become available (e.g., HotpotQA (Yang et al., 2018)), recent years have seen a growing interest in multi-hop question generation. Most recent approaches add heuristically extracted features to the encoder-decoder model, which relies on large-scale training data and can still suffer from error propagation (Yu et al., 2020b; Pan et al., 2020). A recent study (Yu et al., 2020a) which also studies low-resource question generation assumes that a large amount of unanswered multi-hop questions are available, which is also difficult to obtain. We aim to overcome these limitations in this study.

Our study is also related to generative modeling which treats unobserved variables (e.g., features or labels) as latent variables, and approximates the distribution through variational inference (Kingma and Welling, 2014). Generative modeling has been applied to dialogue response generation (Zhao et al., 2019; Huang et al., 2020b; Yang et al., 2020), policy learning (Huang et al., 2019, 2020c),

sentiment analysis (Xu et al., 2017; Li et al., 2019), knowledge retrieval (Lee et al., 2019; Kim et al., 2020a; Su et al., 2021; Tan et al., 2021), and text style transfer (He et al., 2020). While these works focus on utilizing unlabeled data to boost model performance, we aim to unify non-parallel question corpuses to enable joint learning.

6 Conclusions

We proposed a jointly optimized two-phase model named PLAR for low-resource question generation. PLAR effectively utilizes non-parallel single-hop and multi-hop question answering data to perform optimization. We further designed a planning mechanism to guide the generation process of sub-questions so that the generation results are valid to compose a multi-hop question. Experimental results confirm that PLAR achieves better performance compared with the state-of-the-art under various metrics, especially in a question answering based evaluation. For future work, we will explore the heterogeneous multi-hop QG task that requires reasoning beyond plain texts, e.g., tables.

Acknowledgement

This work is supported by Australian Research Council (ARC) Discovery Project DP180102050.

References

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186.

Bhuwan Dhingra, Manzil Zaheer, Vidhisha Balachandran, Graham Neubig, Ruslan Salakhutdinov, and William W. Cohen. 2020. Differentiable reasoning over a virtual knowledge base. In *International Conference on Learning Representations*.

Xinya Du and Claire Cardie. 2018. Harvesting paragraph-level question-answer pairs from Wikipedia. In *Annual Meeting of the Association for Computational Linguistics*, pages 1907–1917.

Yifan Gao, Piji Li, Irwin King, and Michael R. Lyu. 2019. Interconnected question generation with coreference alignment and conversation flow modeling. In *Annual Meeting of the Association for Computational Linguistics*, pages 4853–4862.

Junxian He, Xinyi Wang, Graham Neubig, and Taylor Berg-Kirkpatrick. 2020. A probabilistic formulation of unsupervised text style transfer. In *International Conference on Learning Representations*.

Xinting Huang, Jianzhong Qi, Yu Sun, and Rui Zhang. 2020a. Generalizable and explainable dialogue generation via explicit action learning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3981–3991, Online. Association for Computational Linguistics.

Xinting Huang, Jianzhong Qi, Yu Sun, and Rui Zhang. 2020b. MALA: Cross-domain dialogue generation with action learning. In *AAAI Conference on Artificial Intelligence*, pages 7977–7984.

Xinting Huang, Jianzhong Qi, Yu Sun, and Rui Zhang. 2020c. Semi-supervised dialogue policy learning via stochastic reward estimation. In *Annual Meeting of the Association for Computational Linguistics*, pages 660–670.

Xinting Huang, Jianzhong Qi, Yu Sun, Rui Zhang, and Hai-Tao Zheng. 2019. Carl: Aggregated search with context-aware module embedding learning. In *IJCNN*, pages 101–108. IEEE.

X. Jia, Wenjie Zhou, Xu Sun, and Yunfang Wu. 2020. Eqg-race: Examination-type question generation. In *AAAI Conference on Artificial Intelligence*.

Byeongchang Kim, Jaewoo Ahn, and Gunhee Kim. 2020a. Sequential latent knowledge selection for knowledge-grounded dialogue. In *International Conference on Learning Representations*.

Seokhwan Kim, Mihail Eric, Karthik Gopalakrishnan, Behnam Hedayatnia, Yang Liu, and Dilek Hakkani-Tur. 2020b. Beyond domain APIs: Task-oriented conversational modeling with unstructured knowledge access. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 278–289, 1st virtual meeting. Association for Computational Linguistics.

Yanghoon Kim, Hwanhee Lee, Joongbo Shin, and K. Jung. 2019. Improving neural question generation using answer separation. In *AAAI Conference on Artificial Intelligence*, pages 6602–6609.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational bayes. *CoRR*, abs/1312.6114.

Wei-Jen Ko, Te-yuan Chen, Yiyan Huang, Greg Durrett, and Junyi Jessy Li. 2020. Inquisitive question generation for high level text comprehension. In *Conference on Empirical Methods in Natural Language Processing*, pages 6544–6555.

Dong Bok Lee, Seanie Lee, Woo Tae Jeong, Donghwan Kim, and Sung Ju Hwang. 2020. Generating diverse and consistent QA pairs from contexts with information-maximizing hierarchical conditional VAEs. In *Annual Meeting of the Association for Computational Linguistics*, pages 208–224.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Yitong Li, Timothy Baldwin, and Trevor Cohn. 2019. Semi-supervised stochastic multi-domain learning using variational inference. In *Annual Meeting of the Association for Computational Linguistics*, pages 1923–1934.

Chin-Yew Lin and Eduard Hovy. 2002. Manual and automatic evaluation of summaries. In *ACL-02 Workshop on Automatic Summarization*, page 45–51.

Sewon Min, Victor Zhong, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2019. Multi-hop reading comprehension through question decomposition and rescoring. In *Annual Meeting of the Association for Computational Linguistics*, pages 6097–6109.

N. Mostafazadeh, Chris Brockett, W. Dolan, Michel Galley, Jianfeng Gao, Georgios P. Spithourakis, and Lucy Vanderwende. 2017. Image-grounded conversations: Multimodal context for natural question and response generation. In *International Joint Conference on Natural Language Processing*, pages 462–472.

Aaron van den Oord, Oriol Vinyals, and Kavukcuoglu Koray. 2017. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, pages 6306–6315.

Liangming Pan, Yuxi Xie, Yansong Feng, Tat-Seng Chua, and Min-Yen Kan. 2020. Semantic graphs for generating deep questions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1463–1475.

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

Ethan Perez, Patrick Lewis, Wen-tau Yih, Kyunghyun Cho, and Douwe Kiela. 2020. Unsupervised question decomposition for question answering. In *Conference on Empirical Methods in Natural Language Processing*, pages 8864–8880.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *Conference on Empirical Methods in Natural Language Processing*, pages 2383—-2392.

Sudha Rao and Hal Daumé III. 2019. Answer-based Adversarial Training for Generating Clarification Questions. In *the North American Chapter of the Association for Computational Linguistics*, pages 143–155.

Minjoon Seo, Jinhyuk Lee, Tom Kwiatkowski, Ankur Parikh, Ali Farhadi, and Hannaneh Hajishirzi. 2019. Real-time open-domain question answering with dense-sparse phrase index. In *Annual Meeting of the Association for Computational Linguistics*, pages 4430–4441.

Yixin Su, Rui Zhang, S. Erfani, and Zhenghua Xu. 2021. Detecting beneficial feature interactions for recommender systems. In *AAAI Conference on Artificial Intelligence*, pages 1021–1029.

Yu Sun, Nicholas Jing Yuan, Yingzi Wang, Xing Xie, Kieran McDonald, and Rui Zhang. 2016. Contextual intent tracking for personal assistants. In *SIGKDD*, pages 273–282.

Yu Sun, Nicholas Jing Yuan, Xing Xie, Kieran McDonald, and Rui Zhang. 2017. Collaborative intent prediction with real-time contextual data. *ACM Transactions on Information Systems (TOIS)*, 35(4):1–33.

Yanchao Tan, Carl Yang, Xiangyu Wei, Y. Ma, and X. Zheng. 2021. Multi-facet recommender networks with spherical optimization. *ArXiv*, abs/2103.14866.

Arash Vahdat and J. Kautz. 2020. NVAE: A deep hierarchical variational autoencoder. In *Advances in Neural Information Processing Systems*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Tomer Wolfson, Mor Geva, Ankit Gupta, Matt Gardner, Yoav Goldberg, Daniel Deutch, and Jonathan Berant. 2020. Break it down: A question understanding benchmark. *Transactions of the Association for Computational Linguistics*, 8:183–198.

Weidi Xu, Haoze Sun, Chao Deng, and Ying Tan. 2017. Variational autoencoder for semi-supervised text classification. In *AAAI Conference on Artificial Intelligence*, pages 3358–3364.

Shiquan Yang, Rui Zhang, and Sarah Erfani. 2020. Graphdialog: Integrating graph knowledge into end-to-end task-oriented dialogue systems. In *Conference on Empirical Methods in Natural Language Processing*.

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 *Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380.

Jianxing Yu, Wei Liu, Shuang Qiu, Qinliang Su, Kai Wang, Xiaojun Quan, and Jian Yin. 2020a. Low-resource generation of multi-hop reasoning questions. In *Annual Meeting of the Association for Computational Linguistics*, pages 6729–6739.

Jianxing Yu, Xiaojun Quan, Qinliang Su, and Jian Yin. 2020b. Generating multi-hop reasoning questions to improve machine reading comprehension. *The Web Conference*, pages 281–291.

Qian Yu, Lidong Bing, Qiong Zhang, Wai Lam, and Luo Si. 2020c. Review-based question generation with adaptive instance transfer and augmentation. In *Annual Meeting of the Association for Computational Linguistics*, pages 280–290.

Tiancheng Zhao, Kaige Xie, and Maxine Eskenazi. 2019. Rethinking action spaces for reinforcement learning in end-to-end dialog agents with latent variable models. In *North American Chapter of the Association for Computational Linguistics*, pages 1208–1218.

Yao Zhao, Xiaochuan Ni, Yuanyuan Ding, and Qifa Ke. 2018. Paragraph-level neural question generation with maxout pointer and gated self-attention networks. In *Conference on Empirical Methods in Natural Language Processing*, pages 3901–3910.

A Implementation Details

We use HOTPOTQA split as the original paper (Yang et al., 2018) [4], and use SQuAD v1.1 (Rajpurkar et al., 2016) [5] training set only since single-hop question answering data only involves in the training. We use the BART-base model implementation from huggingface library [6] as the single-hop question generation model and question composing model. We set the batch size to 32 in sentence-level setting and 16 in paragraph-level setting. The models are trained by Adam (Kingma and Ba, 2015) with a learning rate initially set to 3e-5 on NVIDIA GeForce RTX 2080 Ti. We use grid search to find the best hyperparameters for the models based on validation performance, which we use a combination of METEOOR, ROUGE-L and BLEU scores to measure. [7] We set dimensionality of codebook of the planning mechanism (i.e., K) to 100, which is chosen among {50, 75, 100, 150, 200}

B Sub-Question Generation Qualitative Analysis

Table 6 and 7 show question generation results from PLAR and Pipeline model.

C Planning Mechanism Case Study

Table 8 and 9 show the generation results by different sampled planning variables z in the paragraph-level setting. We can see that with different predicted z (denoted by different z_i), PLAR raises different sub-questions and presents different high-level reasoning type. We also find that some planning variable cannot lead to a reasonable multi-hop question, and the prediction of PLAR can well capture the correct plan (denoted by higher $p(z|a, c)$).

Table 6: Sub-Question Generation and Question Composing Examples.

Supportive Evidence	**Paragraph A. Dario Franchitti**
	[1] George Dario Marino Franchitti, MBE (born 19 May 1973), known professionally as Dario Franchitti, is a retired Scottish racing driver. [2] After Franchitti did not secure a single-seater drive in 1995, he was contracted by the AMG team to compete in touring cars in the DTM and its successor — International Touring Car Championship.
	Paragraph B. Mercedes-AMG
	[1] Mercedes-AMG GmbH, commonly known as AMG, is the high performance division of Mercedes-Benz. [2] AMG independently hires engineers, manufactures and customizes Mercedes-Benz AMG vehicles. [3] Mercedes-AMG is headquartered in Affalterbach, Baden Württemberg, Germany.
Groundtruth QA Pair	After he was contracted by the team that is headquartered in Affalterbach, Baden-Württemberg, Germany, Dario Franchitti competed in what series? (Answer: DTM)
PLAR QG results	**Sub-question 1:** Affalterbach Germany is the location of what team?
	Sub-question 2: After contracted by AMG, Dario Franchitti competed in what series?
	Multi-hop question: After he was contracted by the team that is headquartered in Affalterbach, Baden-Württemberg, Germany, Dario Franchitti competed in what series?
Pipeline QG results	**Sub-question 1:** What is headquartered in Affalterbach Baden Germany?
	Sub-question 2: Dario Franchitti competed in what series in 1995?
	Multi-hop question: Dario Franchitti competed in what series in 1995 in Affalterbach Baden Germany?

[4] https://hotpotqa.github.io/
[5] https://rajpurkar.github.io/SQuAD-explorer/
[6] https://huggingface.co/facebook/bart-base
[7] We use the implementations of the metrics as https://github.com/Maluuba/nlg-eval

Table 7: Sub-Question Generation and Question Composing Examples

Supportive Evidence	**Paragraph A. Goran Dragić**
	[1] Goran Dragić (born 6 May 1986) is a Slovenian professional basketball for the Miami Heat of the National Basketball Association (NBA). [2] He plays at both the point guard and shooting guard positions.
	Paragraph B. 2013–14 Phoenix Suns season
	[1] The 2013–14 NBA season was the Phoenix Suns' 46th season in the NBA. [2] When the Suns began the regular season, Goran Dragić, P. J. Tucker, Markieff Morris, and his twin brother Marcus Morris were the only players returning from playing with last season's team (while Channing Frye was still on last season's team, he didn't play any games due to a life-threatening heart ailment he had at the time).
Groundtruth QA Pair	Which team's 2013-2014 season had players including a Slovenian who plays at both the point guard and shooting guard positions? (Answer: the Phoenix Suns)
PLAR QG results	**Sub-question 1:** Which Slovenian player plays at the point guard and shooting guard position?
	Sub-question 2: In 2013 NBA season which team have the player?
	Multi-hop question: In 2013 NBA season which team have the Slovenian player which plays at the point guard and shooting guard position?
Pipeline QG results	**Sub-question 1:** Goran Dragić plays at what positions?
	Sub-question 2: What team begins season with players returning from last season?
	Multi-hop question: Goran Dragić plays at what positions in the team with players returning from last season?

Table 8: Generation Results of Different Sampled Planning Variables

Supportive Evidence	**Paragraph A. Koyaanisqatsi**
	[1] Koyaanisqatsi, also known as Koyaanisqatsi: Life Out of Balance, is a 1982 American experimental film directed by Godfrey Reggio with music composed by Philip Glass and cinematography by Ron Fricke.
	Paragraph B. Mad Hot Ballroom
	[1] Mad Hot Ballroom is a 2005 American documentary film directed and co-produced by Marilyn Agrelo and written and co-produced by Amy Sewell, about a ballroom dance program in the New York City Department of Education, the New York City public school system for fifth graders. [2] Several styles of dance are shown in the film, such as tango, foxtrot, swing, rumba and merengue.
Groundtruth QA Pair	Which film was created more recently, Koyaanisqatsi or Mad Hot Ballroom? (Answer: Mad Hot Ballroom)
PLAR w/ Planning z_1 $p(z_1\|c,a)$=0.52	**Sub-question 1:** Which year is film Koyaanisqatsi created?
	Sub-question 2: Which year is film Mad Hot Ballroom created?
	Multi-hop question: Film Koyaanisqatsi and Mad Hot Ballroom, which is created later?
PLAR w/ Planning z_2 $p(z_2\|c,a)$=0.33	**Sub-question 1:** What is the name of file directed by Marilyn Agrelo?
	Sub-question 2: What 1982 experimental film Godfrey Reggio directed?
	Multi-hop question: Are Koyaanisqatsi and Mad Hot Ballroom by the same director?
PLAR w/ Planning z_3 $p(z_3\|c,a)$=0.10	**Sub-question 1:** What movie has music by Philip Glass?
	Sub-question 2: What movie shows dance styles such as tango, foxtrot?
	Multi-hop question: What movie has music by Philip Glass and dance styles such as tango, foxtrot?

Table 9: Generation Results of Different Sampled Planning Variables

Supportive Evidence	**Paragraph A. Force India VJM10**
	[1] The Force India VJM10 is a Formula One racing car designed and constructed by Force India to compete during the 2017 Formula One season. [2] The car is driven by Sergio Pérez and Esteban Ocon, who joined the team after Nico Hülkenberg left the team at the end of the season.
	Paragraph B. Esteban Ocon
	[1] Esteban Ocon (born 17 September 1996) is a French racing driver who currently drives in Formula One for Force India. [2] He made his Formula One debut for Manor Racing in the 2016 Belgian Grand Prix, replacing Rio Haryanto. [3] Ocon is part of the Mercedes-Benz driver development programme.
Groundtruth QA Pair	Force India VJM10 is a Formula One racing car previous driven by Nico Hülkenberg, and is now driven by which driver born 17 September 1996? (Answer: Esteban Ocon)
PLAR w/ Planning z_1 $p(z_1\|c,a)$=0.71	**Sub-question 1:** who drives Force India VJM10, a Formula One racing car previous driven by Nico Hülkenberg?
	Sub-question 2: which driver is born 17 September 1996?
	Multi-hop question: who is born 17 September 1996 and drives Force India VJM10, a Formula One racing car previous driven by Nico Hülkenberg?
PLAR w/ Planning z_2 $p(z_2\|c,a)$=0.11	**Sub-question 1:** Who currently drives in Fomula One for Force India?
	Sub-question 2: Who joined Force India after Nico Hülkenberg left?
	Multi-hop question: Who joined and drives for Force India after Nico Hülkenberg left?