Explicit over Implict: Explicit Diversity Conditions for Effective Question Answer Generation

Vikas Yadav, Hyuk Joon Kwon, Vijay Srinivasan, Hongxia Jin

Samsung Research America, Mountain View

vikasy.zona@gmail.com, {l.kwon, v.srinivasan, hongxia.jin}@samsung.com

Abstract

Question Answer Generation (QAG) is an effective data augmentation technique to improve the accuracy of question answering systems, especially in low-resource domains. While recent pretrained and large language model-based QAG methods have made substantial progress, they face the critical issue of redundant QA pair generation, affecting downstream QA systems. Implicit diversity techniques such as sampling and diverse beam search are proven effective solutions but often yield smaller diversity. We present explicit diversity conditions for QAG, focusing on spatial aspects, question types, and entities, substantially increasing diversity in QA generation. Our work emphasizes the need of explicit diversity conditions for generating diverse question-answer synthetic data by showing significant improvements in downstream QA task over existing widely adopted implicit diversity techniques. In particular, generated QA pairs from explicit diversity conditions when used to train the downstream QA model results in an average 4.1% exact match and 4.5% F1 improvement over QAG from implicit sampling techniques on SQuAD$_{DU}$. Our work emphasizes the need for explicit diversity conditions even more in low-resource datasets (SubjQA), where average downstream QA performance improvements are 12% EM.

Keywords: Question Answer Generation (QAG), Explicit conditions for diverse QAG, Coverage of QAG

1. Introduction

Annotating QA pairs is costly, tedious, and constrained to annotators' limited coverage of the input document which often leads to lower QA performance in low resource domains (Rajpurkar et al., 2016; Bartolo et al., 2020; Yadav et al., 2019). Recent QAG methods, particularly neural pretrained language models (PLM) and large language models (LLM), have generated high-quality synthetic QA pairs leading to strong downstream QA performance (Du and Cardie, 2018; Puri et al., 2020a; Stasaski et al., 2021). It is reported that even these prominent neural QAG methods suffer from repeated redundant generation, even after utilizing several implicit techniques for diverse generations such as nucleus, topK sampling, and diverse decoding methods (Shao et al., 2017; Sultan et al., 2020). Our work evaluates diversity of such widely adopted implicit techniques for QAG and show the QA generations to be still largely redundant, affecting downstream QA performance. We conjecture that artifacts in human annotations of the training data leads to QAG redundancy. For example, 71% of the questions in the benchmark QAG dataset SQuAD$_{DU}$ are annotated from the first half of the document, and 73% of the questions are of the type *who, how, what, and why*. As shown in fig. 1, human annotators have annotated QA pairs only from the top and 4/5th position of the passage and only *what* and *how* questions. Training on such skewed dataset may overfit neural QAG methods on numerous annotator artifacts, thus reducing diversification effectiveness of implicit sampling techniques.

Our work focuses on explicit diversity conditions where we present three types of explicit prompts, conditioning QAG on (1) various positions (*POS*) within the input document from where QA pairs are generated, (2) 8 types of *WH* questions for generating questions of different types, and (3) questions based on different named entities (*ENT*). As shown in upper block of fig. 1, these explicit diversity conditions are concatenated as prompts to the input document for diverse QA generation. These explicit conditions can also be easily combined to one another for jointly prompting QAG models, especially the LLM based ones where we observed the best downstream QA performance (§4). Our work primarily focuses on establishing the importance of adding diversity conditions explicitly over the widely adopted implicit sampling techniques. The clear benefits of explicit prompting based QAG are highlighted with improved downstream QA performance (§4) and coverage of diverse information (§5) from the input document. Our key contributions are:

(1) We study diversity of implicit sampling techniques and compare them with several explicit diversity conditions for QAG. The synthetic QA pairs generated from our explicit diversity conditions significantly improve the downstream QA performance outperforming implicit sampling techniques by 4.1% EM and 4.5% F1 on widely studied SQuAD$_{DU}$ dataset (Du et al., 2017). The improvements from our explicit conditions drastically exceed in the multi-domain low resource SubjQA dataset (Ushio et al., 2022) with improvements of 12% F1 score.

(2) Our explicit diversity prompts show substantial diversity improvements, resulting in only 30% token overlap among generated QA pairs from the input document, compared to the 64% overlap in
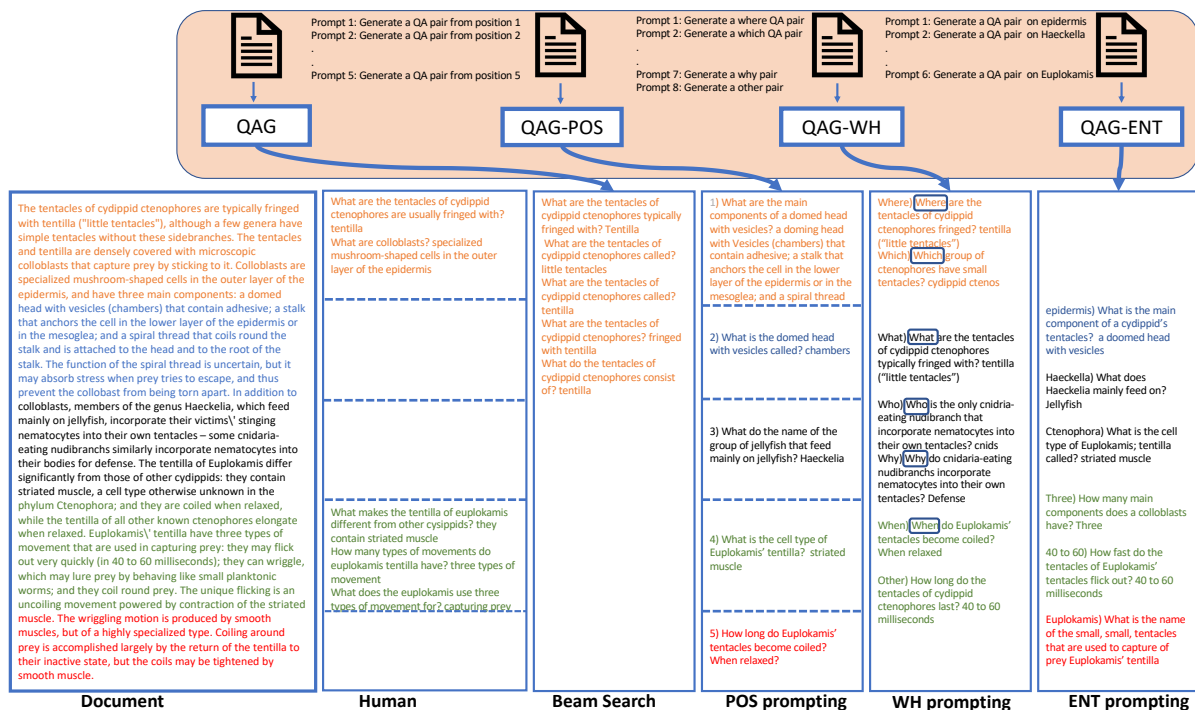
QAG   **QAG-POS**   **QAG-WH**   **QAG-ENT**

Document

The tentacles of cydippid ctenophores are typically fringed with tentilla ("little tentacles"), although a few genera have simple tentacles without these sidebranches. The tentacles and tentilla are densely covered with microscopic colloblasts that capture prey by sticking to it. Colloblasts are specialized mushroom-shaped cells in the outer layer of the epidermis, and have three main components: a domed head with vesicles (chambers) that contain adhesive; a stalk that anchors the cell in the lower layer of the epidermis or in the mesoglea; and a spiral thread that coils round the stalk and is attached to the head and to the root of the stalk. The function of the spiral thread is uncertain, but it may absorb stress when prey tries to escape, and thus prevent the colloblast from being torn apart. In addition to colloblasts, members of the genus Haeckelia, which feed mainly on jellyfish, incorporate their victims\' stinging nematocysts into their own tentacles – some cnidaria-eating nudibranchs similarly incorporate nematocytes into their bodies for defense. The tentilla of Euplokamis differ significantly from those of other cydippids: they contain striated muscle, a cell type otherwise unknown in the phylum Ctenophora; and they are coiled when relaxed, while the tentilla of all other known ctenophores elongate when relaxed. Euplokamis\' tentilla have three types of movement that are used in capturing prey: they may flick out very quickly (in 40 to 60 milliseconds); they can wriggle, which may lure prey by behaving like small planktonic worms; and they coil round prey. The unique flicking is an uncoiling movement powered by contraction of the striated muscle. The wriggling motion is produced by smooth muscles, but of a highly specialized type. Coiling around prey is accomplished largely by the return of the tentilla to their inactive state, but the coils may be tightened by smooth muscle.

Human

What are the tentacles of cydippid ctenophores are usually fringed with? tentilla

What are colloblasts? specialized mushroom-shaped cells in the outer layer of the epidermis

---

What makes the tentilla of euplokamis different from other cysippids? they contain striated muscle

How many types of movements do euplokamis tentilla have? three types of movement

What does the euplokamis use three types of movement for? capturing prey

Beam Search

What are the tentacles of cydippid ctenophores typically fringed with? Tentilla

What are the tentacles of cydippid ctenophores called? little tentacles

What are the tentacles of cydippid ctenophores called? tentilla

What are the tentacles of cydippid ctenophores? fringed with tentilla

What do the tentacles of cydippid ctenophores consist of? tentilla

POS prompting

1) What are the main components of a domed head with vesicles? a doming head with Vesicles (chambers) that contain adhesive; a stalk that anchors the cell in the lower layer of the epidermis or in the mesoglea; and a spiral thread

2) What is the domed head with vesicles called? chambers

3) What do the name of the group of jellyfish that feed mainly on jellyfish? Haeckelia

4) What is the cell type of Euplokamis' tentilla? striated muscle

5) How long do Euplokamis' tentacles become coiled? When relaxed?

WH prompting

Where) Where are the tentacles of cydippid ctenophores fringed? tentilla ("little tentacles")

Which) Which group of ctenophores have small tentacles? cydippid ctenos

What) What are the tentacles of cydippid ctenophores typically fringed with? tentilla ("little tentacles")

Who) Who is the only cnidaria-eating nudibranch that incorporate nematocytes into their own tentacles? cnids

Why) Why do cnidaria-eating nudibranchs incorporate nematocytes into their own tentacles? Defense

When) When do Euplokamis' tentacles become coiled? When relaxed

Other) How long do the tentacles of cydippid ctenophores last? 40 to 60 milliseconds

ENT prompting

epidermis) What is the main component of a cydippid's tentacles? a doomed head with vesicles

Haeckela) What does Haeckelia mainly feed on? Jellyfish

Ctenophora) What is the cell type of Euplokamis; tentilla called? striated muscle

Three) How many main components does a colloblasts have? Three

40 to 60) How fast do the tentacles of Euplokamis' tentacles flick out? 40 to 60 milliseconds

Euplokamis) What is the name of the small, small, tentacles that are used to capture of prey Euplokamis' tentilla

Figure 1: A sample input passage and QA pairs generated by human annotators, nucleus sampling based beam search and our explicit diversity prompting techniques. Different colors in the document text depict the 5 different positions. QA pairs from specific positions are depicted in the same font color and *WH* question types are indicated in blue bounding boxes. Example of each explicit diversity prompts are shown in the top block.

QA pairs from implicit sampling-based QAG. The coverage of information from the input document in terms of position, question type, and named entity attributes is also considerably higher in generated QA pairs from explicit diversity prompting over implicit sampling techniques.

2. Related Work

Recent studies have highlighted redundancy in neural-QAG approaches and while some widely adopted diverse sampling and beam decoding methods (Sultan et al., 2020; Holtzman et al., 2019; Vijayakumar et al., 2018) have shown improvements, these implicit techniques only moderately enhance the diversity of generation (see table 3). Furthermore, implicit sampling techniques lack precise control of QAG for accessing specific information from the input document. For example, QAG models using nucleus sampling or diverse decoding would still generate QA pairs from a random position of the document or of random WH question type. In contrast, our explicit prompting techniques offer high control over QA generation, allowing selection from specific positions or named entities in the input document, and the types of questions (shown in last 3 columns of fig. 1).

Many previous QAG methods can be broadly categorized as either explicit or implicit techniques. For instance, Zhou et al. (2019) is analogous to our explicit WH-type QAG model, while answer selector modules (Yao et al., 2022; Back et al., 2021; Puri et al., 2020b), which select answer spans to con-

dition QG, are analogous to our entity-conditioned QAG method. On the other hand, sampling, beam search and additional embedding (Lee et al., 2020) based approaches can be grouped under implicit diversity conditions. From these, we mainly focus on widely adopted sampling and diverse decoding methods as implicit diversity baselines(§3.3). Our work primarily focuses on comparing these two broad directions of explicit versus implicit diversity methods by showing their impact on diverse generation, downstream QA performance, and information coverage from the input document. We show experiments on the standard question generation benchmark - QGbench (Ushio et al., 2022). QGbench authors highlighted higher performance from pretrained-LMs over RNN based models. For fair comparisons, we implemented both explicit and implicit sampling techniques on the same base PLM (BART (Lewis et al., 2020)) and LLM (LLaMa-7B (Touvron et al., 2023)) models. Further, Ushio et al. (2023) showed end-to-end QAG as the best setting where both question and answer are generated as a single output, in a single generation step. We use the same setting throughout our experiments.

3. Approach

The task of QAG is to generate QA pairs given an input document. Formally, given a document D which contains M tokens, $D = (d_1, ..., d_M)$ and i^{th} QA pair has t number of tokens i.e., $qa^i = (qa^i_1, ..., qa^i_t)$, the task is formulated as a condi-

tional sequence generation at the token level i.e., $P(qa_k^i|qa_1^i...qa_{k-1}^i, d_1, ..., d_M)$. We model the conditional probability $P(qa|D)$ using 1) BART, a PLM that achieves best results on QGbench and 2) LLaMa-7B, a decoder only LLM. Our three explicit diversity conditions are described below.

3.1. Explicit diversity prompts

POS prompting: We consider 5 splits of the input document based on its total word count. For example, if a document contains 400 tokens, each split will cover 80 tokens each. QAG$_{POS}$ is then conditioned on positions of each of the 5 splits, thus encouraging generation of QA pairs from 5 different positions[1]. In particular, we explicitly prompt the QAG$_{POS}$ model to generate qa from pos position of the document where $pos \in 1, 2, 3, 4, 5$.

$$qa_{pos} \sim P(qa|D, pos) \qquad (1)$$

For example, to generate a QA pair from the 2nd position of the document (shown by blue font in the 1st column of fig. 1), we prompt: "Generate a QA pair from position 2" to our QAG$_{pos}$ model. Splitting of the input document based on word count is a bit rigid segmentation. However, we observed that even with such rough segmentation, QAG$_{pos}$ model is able to learn approximate alignment between position of the input document and its corresponding generated QA pair. During training, we use the start offset of the human annotated answer to determine which position split the QA pair was annotated from. During inference, we generate 5 QA pairs from all the 5 different positions of the document.

WH prompting: Similar to POS prompting, we condition on the wh question type where $wh \in$ {*"where", "which", "when", "what", "who", "how", "why"*} to encourage the QAG$_{WH}$ model to generate different types of questions.

$$qa_{wh} \sim P(qa|D, wh) \qquad (2)$$

During training of QAG$_{WH}$, we use the wh type from human annotated questions and during inference, we simply generate QA pairs by conditioning on all 7 wh types. If the annotator's question did not have any wh type, then we consider it as *"other"*. As shown in 2nd last column of fig. 1, prompting QAG$_{WH}$ with wh generates diverse QA pairs with different question type.

ENT prompting: QAG$_{ENT}$ is conditioned on named entities in the input document to generate entity-specific QA pairs. During training, we select named entities present in the human annotated QA pairs and the input document, identified using the SpaCy NER tagger with 18 entity classes from OntoNotes(Weischedel et al.). During inference,

we split the document into individual sentences and select the longest named entity from each sentence to use in the prompt. The named entity-conditioned prompt, along with the input document, generates a QA pair for that specific entity. As shown in fig. 1, QAG$_{ENT}$ generates diverse QA pairs by conditioning on different entities from the input document.

3.2. Combined prompts

Our three base diversity prompts can be rigid sometimes; for example, a specific WH question may not be feasible for a particular document. To address this issue, we propose a two step process, with the first step being wh question type prediction given a position or entity. For example, given $pos = 2$, a trained wh predictor model predicts "what" type of question in the 1st step. In the 2nd step, QAG model generates a "what" type question from the 2nd position of the document. This two step process of combining wh type with position and entity diversity conditions is explained below.

- **Position-based question type generator**: We train a separate BART model to generate a list of relevant WH-types (or 'none' if no QA is possible) for a specific position in the input document. Then, we generate QA pairs conditioned on both the specified position and the predicted WH types.

$$wh \sim P(wh|D, pos) \qquad (3)$$
$$qa_{pos,wh} \sim P(qa|D, pos, wh) \qquad (4)$$

- **Entity-based question type generator**: Similarly, we predict potential wh question types for the selected entity from the input document. We then generate QA pairs given the selected entity and the predicted WH types.

$$wh \sim P(wh|D, ent) \qquad (5)$$
$$qa_{ent,wh} \sim P(qa|D, ent, wh) \qquad (6)$$

Please note that this 2 step process for combing different explicit diversity conditions is required only for BART-QAG. As LLMs can follow instructions and generate long sequences (Wei et al., 2022), a single prompt for combining two explicit conditions - *"Generate N questions of different question type from different positions"* is given to the LLaMa-7B QAG model. This single prompt based combining of explicit conditions for QAG is referred to as $Combined$ in table 1.

3.3. Implicit Sampling and Decoding

We considered four widely adopted decoding techniques as implicit diversity baselines: nucleus sampling, top_k sampling, beam search with sampling, and diverse decoding (Sultan et al., 2020; Holtzman et al., 2019; Vijayakumar et al., 2016, 2018). In addition, we also assessed these sampling techniques in combination with our explicit diversity

---

[1]We tried different number of positions from the document $\in 2, 5, 10$ and found the best QAG with 5 positions.

conditions. Our position and entity prompt conditioned QAG models performed consistently better with diverse decoding, while WH prompting showed higher downstream QA performance with nucleus sampling.

Hyperparameters: We conducted our experiments with the PyTorch implementation of BART and LLaMa from the Hugging Face Transformers library (Wolf et al., 2020). For training BART-QAG, the final hyperparameters were epoch=4, batch size=16, learning rate=3e-5, and adam optimizer on V100 GPUs. The remaining hyperparameters were used as default suggested in (Ushio et al., 2022). We used 4 A100 GPUs to fine-tune our LLaMa-QAG models using following hyperparameters: batch size=4, learning rate=2e-5, epoch=3, and float32=True. For training the BERT-large-uncased-wwm QA model, the final hyperparameters were epoch=2, learning rate=3e-5, seq length=384, batch size=8, and stride=128

4. Evaluation

We evaluate the impact of our explicit diversity prompts on downstream QA task on the standard datasets from QGbench (i) SQuAD$_{DU}$(Du et al., 2017) (ii) and low-resource multi-domain SubjQA (Bjerva et al., 2020) which has less than 150 annotations. We trained a BERT-large-uncased-wwm based QA model (Devlin et al., 2019) over synthetic QA pairs generated from our explicit conditions and implicit sampling techniques. Each experiments is run 5 times and average QA performance on the SQuAD$_{DU}$ test split is reported in table 1. We report both the F1 and exact match (EM) using the standard SQuAD evaluation script (Rajpurkar et al., 2016).

(1) Downstream QA performance: QA model trained on data from explicit diversity prompted QAG models (row-block 2 of table 1) achieve, on average, 4.1% higher EM and 4.5% higher F1 scores compared to the implicit sampling techniques (row-block 1 of table 1) in our BART-QAG methods. These empirical evidences highlight the importance of explicit diversity conditioning for effective QAG. Performances improve further (row-block 3) when diversity conditions are combined in a learned setting (§3.2), suggesting the need for a learned module to capture complex relationships between the three explicit diversity conditions. Interestingly, combining multiple explicit conditions in a single prompt for LLaMa-QAG (denoted by *Combined* in table 1) results in the best downstream QA performance.

(2) BART vs LLaMa - QA pairs generated from LLaMa-QAG consistently lead to better downstream performance than BART-QAG. As expected,

SRC	I/E	Approach	BART Orig Size		LLaMa-7B Orig Size	
			EM	F1	EM	F1
Syn	I	Greedy	64.76	76.66	71.41	83.26
Syn	I	Nucl (0.95)	64.44	77.15	71.92	83.53
Syn	I	Nucl+TopK	64.17	76.47	72.08	83.71
Syn	I	DiverseDec.	65.21	77.37	71.87	83.66
Syn	E	WH Prompt	67.25	79.60	72.97	84.46
Syn	E	POS Prompt	69.62	81.49	72.74	84.25
Syn	E	ENT Prompt	69.31	81.80	72.59	84.21
Syn		POS->WH	71.77	83.30	-	-
Syn	EL	ENT->WH	70.46	81.78	-	-
Syn		Combined	-	-	**73.29**	**84.76**
H+Syn		SQ$_{dev}$+WH	74.30	85.62	75.11	86.42
H+Syn		SQ$_{dev}$+POS	74.53	85.61	**75.76**	**87.15**
H+Syn		SQ$_{dev}$+ENT	73.17	85.01	75.59	87.06
H		SQ$_{dev}$	EM=74.08		F1=85.19	

Table 1: Downstream QA performance on the QG-bench SQuAD DU test dataset. We use top$_p$=0.95 and topK=30.The third-row block settings refer to the learned combination of diversity conditions (§3.2) where the first prompt predicts the second potential diversity prompt (separated by ->). I, E, and EL in the 2nd column stand for implicit, explicit, and learned explicit conditons respectively. SQ refers to SQuAD$_{DU}$ dev split. Nucl and DiverseDec are short form for Nucleus and DiverseDecoding. The Orig Size indicates that the synthetic data size matches the original training size of SQuAD DU dataset of 10570 QA pairs. The eval dataset for all the rows is the SQuAD DU test split which contains 11877 QA pairs. H and Syn refers to human annotated and synthetic QA dataset.

Data	Eval	Hum	BART-QAG			
			NS	POS	WH	ENT
Books	EM	6.3	7.9	14.7	11.6	20.0
T,E=92,191	F1	20.3	25.9	29.4	30.1	37.9
Electronics	EM	15.2	16.4	23.6	27.9	25.7
T,E=99,238	F1	34.3	33.6	44.4	47.5	47.5
Grocery	EM	14.6	16.5	16.2	0.0	15.4
T,E=101,379	F1	31.9	32.0	31.2	16.1	31.1
Movies	EM	13.6	15.6	23.4	27.9	25.3
T,E=101,154	F1	30.2	30.2	36.9	41.3	39.3
Restaurant	EM	8.2	0.0	6.7	12.7	26.1
T,E=129,136	F1	23.9	7.1	20.3	25.9	40.3

Table 2: Downstream QA performance on the SubjQA test dataset of QG-bench. NS refers to nucleus sampling. Numbers in bold represent the best performance in each column. T and E under each domain refer to the number of QA pairs in training and evaluation split respectively. Please note that BART-QAG model geenrates the same number of QA pairs as annotated in Human (Hum) sets in each domain.

the improvements are smaller from explicit conditions in LLaMa-QAG because of their extensive pretraining leading to more qualitative generations. LLaMa-QAG synthetic QA data from explicit conditions almost matches performance from human annotated SQuAD$_{DU}$ dataset (within 0.4% F1). Interestingly, just appending QA pairs from explicit conditioned LLaMa-QAG to human annotated SQuAD$_{DU}$ leads to 2% F1 improvement (row block 4), resulting in best performance of 87.2% F1 in downstream QA task. This highlights the benefits of combining high-quality diverse synthetic data to existing human annotated QA pairs. Although average improvements in downstream QA tasks from explicit diversity prompts are smaller in LLaMa-QAG, the generated QA pairs still have higher coverage and diversity compared to implicit sampling techniques (discussed in §5).

(3) Low resource QAG - In table 2, we observed substantially higher performance improvements with our explicit diversity-conditioned BART-QAG

Analysis	Overlap	Coverage			Time
		POS	WH	ENT	(ms)
Greedy	63.07	36.84	31.33	32.18	223.1
Nucl (0.95)	57.44	57.15	45.59	29.80	372.1
Nucl+TopK	59.93	58.62	48.23	30.21	451.4
DiverseDec	46.85	49.83	42.76	35.38	388.2
POS Prompt	36.10	**77.56**	34.62	50.62	231.5
WH Prompt	**30.67**	60.41	**97.81**	48.06	218.7
ENT Prompt	34.59	75.89	55.34	**63.90**	227.9
Human	**28.04**	65.82	56.32	44.96	-

Table 3: Pairwise lexical overlap between generated QA tokens, their coverage, and average generation time for 5 QA pairs from SQuAD$_{DU}$.

on the SubjQA datasets. Particularly, synthetic data from explicit diversity-conditioned BART-QAG resulted in a 7% EM and 10% F1 improvement over implicit nucleus sampling based QA data. Interestingly, explicit-conditioned QA pairs lead to on-par or higher performance when compared to small-sized human annotated data of SubjQA. Thus, emphasizing the importance of explicit diversity conditions even more in low-resourced domains.

5. Overlap and Coverage Analyses

We compute the lexical token overlap between the generated QA pairs for each document. For this analysis, we generated 5 questions with each approach and report the average pairwise token lexical overlap between all $\binom{5}{2}$ QA pairs over SQuAD$_{DU}$ dev split. As shown in table 3, there is a substantially higher average token overlap of 63.1% between QA pairs generated by greedy beam search clearly highlighting the diversity problem. Nucleus sampling and diverse decoding have comparatively lower overlap ratios (57.4 and 49.8) but are still substantially higher than our techniques suggesting the need of explicit diversity conditioning. WH explicit prompting results in the lowest average token overlap of just 30.7 indicating its effectiveness in diverse QAG. It is worth noting that the overlap of human annotated QA pairs were low because there were ≤ 2 QA pairs annotated for majority of the input documents.

We also compute the average lexical coverage of the 5 generated QA pairs by assessing answer text position (POS), entity and *wh* in question text (denoted by ENT and WH in §5). For example, we compute answer position coverage of the generated QA pairs from position 1 to 5. If generated QA pairs have answers only in 4 of the 5 splits, POS coverage will be 80%. QA pairs generated by our explicit diversity-conditioned methods have substantially higher coverage compared to all the implicit sampling baselines. Unsurprisingly, BART$_{POS}$, BART$_{WH}$, and BART$_{ENT}$ have the highest average lexical coverage of spatial, *wh* question type, and named entity in the input document respectively. We also calculate the average generation time of 5 QA pairs per input document (last column of Table 3) highlighting explicit diversity prompts are

also much faster than selecting multiple beams and other diverse decoding techniques.

Conclusion: We presented a detailed study of implicit versus explicit conditioning techniques for diverse QA generation, highlighting lack of diversity in generations from implicit techniques. Our work empirically shows the clear benefits of explicit diversity conditions with substantial improvements in diverse generations, downstream QA task, and information coverage from the input document. We also show that the concatenation of explicit conditioned based diverse synthetic QA pairs to human annotated datasets leads to further improvement in downstream QA performance. Overall, our presented findings suggest the need of utilizing more of explicit diversity conditions over the existing popular diversity sampling techniques, especially in low resource settings.

6. Future Work

We focus on the standard and more mainstream QAG task from QGBench but our proposed techniques can be easily extended to other complex QA tasks such as multi-hop QA (Yadav et al., 2020). Similarly, our explicit diversity techniques can be extended to other text generation tasks such as conversational QA, dialogue, answer summarization etc (Reddy et al., 2019; Wu et al., 2022).

In case of position diversity of generated QA pairs, input documents can be longer (or shorter). Although we had tried splits $\in \{2,5,10\}$, the number of position splits can be variably selected depending on its length in future works. In §5, we studied diversity in terms of overlap and coverage of information via simple lexical matching. As future work, the embedding representations of the generated questions throughout QAG model layers can also be evaluated to further understand the effects of training with explicit diversity conditions (Yadav et al., 2021).

7. Ethical consideration

We simply utilize benchmark datasets from QGbench and existing PLM and LLM models like BART and LLaMa-7B. Our presented methodologies and comparitive studies do not induce any biased or harmful content. We believe, similar to the other LLM and PLM based systems, risks depends on the underlying LLM from its pretraining. A careful selection of input documents for QAG and unbiased LLMs or PLMs would ensure safe QA generations from either explicit or implicit techniques. To the best of our knowledge and review of nearly 200 generated QA pairs, we did not find any harmful or biased QA generations.

8. Bibliographical References

Seohyun Back, Akhil Kedia, Sai Chetan Chinthakindi, Haejun Lee, and Jaegul Choo. 2021. Learning to generate questions by learning to recover answer-containing sentences. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1516–1529, Online. Association for Computational Linguistics.

Max Bartolo, Alastair Roberts, Johannes Welbl, Sebastian Riedel, and Pontus Stenetorp. 2020. Beat the AI: Investigating adversarial human annotation for reading comprehension. *Transactions of the Association for Computational Linguistics*, 8:662–678.

Johannes Bjerva, Nikita Bhutani, Behzad Golshan, Wang-Chiew Tan, and Isabelle Augenstein. 2020. Subjqa: A dataset for subjectivity and review comprehension. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5480–5494.

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 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Xinya Du and Claire Cardie. 2018. Harvesting paragraph-level question-answer pairs from Wikipedia. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1907–1917, Melbourne, Australia. Association for Computational Linguistics.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1342–1352, Vancouver, Canada. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

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 *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 208–224, Online. Association for Computational Linguistics.

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 *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Raul Puri, Ryan Spring, Mohammad Shoeybi, Mostofa Patwary, and Bryan Catanzaro. 2020a. Training question answering models from synthetic data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5811–5826, Online. Association for Computational Linguistics.

Raul Puri, Ryan Spring, Mohammad Shoeybi, Mostofa Patwary, and Bryan Catanzaro. 2020b. Training question answering models from synthetic data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5811–5826.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Yuanlong Shao, Stephan Gouws, Denny Britz, Anna Goldie, Brian Strope, and Ray Kurzweil. 2017. Generating high-quality and informative conversation responses with sequence-to-sequence models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2210–2219, Copenhagen, Denmark. Association for Computational Linguistics.

Katherine Stasaski, Manav Rathod, Tony Tu, Yunfang Xiao, and Marti A. Hearst. 2021. Automatically generating cause-and-effect questions from passages. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 158–170, Online. Association for Computational Linguistics.

Md Arafat Sultan, Shubham Chandel, Ramón Fernandez Astudillo, and Vittorio Castelli. 2020. On the importance of diversity in question generation for QA. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5651–5656, Online. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Asahi Ushio, Fernando Alva-Manchego, and Jose Camacho-Collados. 2022. Generative language models for paragraph-level question generation. *arXiv preprint arXiv:2210.03992*.

Asahi Ushio, Fernando Alva-Manchego, and Jose Camacho-Collados. 2023. An empirical comparison of LM-based question and answer generation methods. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 14262–14272, Toronto, Canada. Association for Computational Linguistics.

Ashwin Vijayakumar, Michael Cogswell, Ramprasaath Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2018. Diverse beam search for improved description of complex scenes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2016. Diverse beam search: Decoding diverse solutions from neural sequence models. *arXiv preprint arXiv:1610.02424*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Ralph Weischedel, Sameer Pradhan, Lance Ramshaw, et al. Ontonotes release 4.0.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Chien-Sheng Wu, Andrea Madotto, Wenhao Liu, Pascale Fung, and Caiming Xiong. 2022. QA-Conv: Question answering on informative conversations. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5389–5411, Dublin, Ireland. Association for Computational Linguistics.

Vikas Yadav, Steven Bethard, and Mihai Surdeanu. 2019. Quick and (not so) dirty: Unsupervised selection of justification sentences for multi-hop question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2578–2589.

Vikas Yadav, Steven Bethard, and Mihai Surdeanu. 2020. Unsupervised alignment-based iterative evidence retrieval for multi-hop question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4514–4525.

Vikas Yadav, Steven Bethard, and Mihai Surdeanu. 2021. If you want to go far go together: unsupervised joint candidate evidence retrieval for multi-hop question answering. In *Proceedings of the 2021 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 4571–4581.

Bingsheng Yao, Dakuo Wang, Tongshuang Wu, Zheng Zhang, Toby Li, Mo Yu, and Ying Xu. 2022. It is AI's turn to ask humans a question: Question-answer pair generation for children's story books. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 731–744, Dublin, Ireland. Association for Computational Linguistics.

Wenjie Zhou, Minghua Zhang, and Yunfang Wu. 2019. Question-type driven question generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6032–6037.