ECAI 2023
K. Gal et al. (Eds.)

doi:10.3233/FAIA230539

Uniform Training and Marginal Decoding for Multi-Reference Question-Answer Generation

Svitlana Vakulenko[a;*], **Bill Byrne**[ab] **and Adrià de Gispert**[a]

[a]Amazon Alexa AI
[b]University of Cambridge

Abstract. Question generation is an important task that helps to improve question answering performance and augment search interfaces with possible suggested questions. While multiple approaches have been proposed for this task, none addresses the goal of generating a diverse set of questions given the same input context. The main reason for this is the lack of multi-reference datasets for training such models. We propose to bridge this gap by seeding a baseline question generation model with named entities as candidate answers. This allows us to automatically synthesize an unlimited number of question-answer pairs. We then propose an approach designed to leverage such multi-reference annotations, and demonstrate its advantages over the standard training and decoding strategies used in question generation. An experimental evaluation on synthetic, as well as manually annotated data shows that our approach can be used in creating a single generative model that produces a diverse set of question-answer pairs per input sentence.

1 Introduction

A significant limitation of current question generation models is that they are trained to produce only a single question based on a given input context [19, 13]. This is a shortcoming in that many sentences, particularly in the news or science domains, are long and information-dense. These sentences can easily give rise to multiple interesting and useful questions. While standard question generation methods can produce multiple questions, these are usually paraphrases of the same question rather than semantically distinct questions.

Improved diversity of generated questions was identified as an important research direction in a literature review covering the recent research on question-answer generation (QAG) [19]. The authors note that the standard approaches primarily focus on generating one question as a 1-to-1 mapping problem and consequently, such methods fail to generate multiple diverse questions. The major challenge they see with respect to diverse QAG is in *identifying different question-worthy context words dynamically* and actively controlling the question generation process.

Automatically generated questions are useful for fine-tuning question answering (QA) models [17, 14] and augmenting document collections to improve matching [10, 1, 16, 3]. In supervised learning, richer training sets are likely to lead to more robust and diverse QA models. Generating a diverse set of questions was also shown to improve performance on non-QA downstream tasks that require general passage representations, such as paraphrase detection, named entity recognition and sentiment analysis [7].

To enable diverse QAG, we posit the task of generating a set of questions from an input context, such as a single sentence or a paragraph of text. We are interested in generating a diverse set of questions that focus on different aspects of the input text rather than mere paraphrases of the same question. To achieve this, we make use of the simple observation that different questions tend to have different answers. Using different answer spans extracted from the input text can ensure generation of a diverse set of questions.

To the best of our knowledge, we are the first to propose an approach for extending a state-of-the-art QAG model to a multi-reference setting, i.e., a training scenario in which multiple correct output sequences are available for matching the same input sequence. We also streamlined training and evaluation of our approach by generating a multi-reference QA dataset centered around named entities.

Named entities, such as names, locations and dates, are suitable targets for fact-based QA, and related work on question generation often generates questions with answers that are named entities [4, 18]. When there are multiple entities per sentence, our approach can generate automatically a dataset with multiple distinct QA pairs. We will use a combination of an off-the-shelf named entity recognition (NER) model and a standard QAG model to produce synthetic datasets for our experiments.

The goal of our work is to develop variations in model training and decoding to account for multiple QA pairs in the ground truth. The NER model here is used as a proxy for generating a multi-reference dataset as a proof-of-concept for our approach. More question-answer pairs can be obtained by conducting manual annotations or using other models, such as keyword extraction and summarization. Even simple heuristics such as noun phrase extractors can help to identify answer spans that potentially lead to useful questions.

While most of the related work focuses on generating questions given a passage [10, 12], we take a more ambitious goal of generating multiple diverse QA pairs given a single sentence. It is more difficult since the search space is much smaller and the likelihood of generating a duplicate is much higher. None of the existing QAG datasets, such as SQuAD [12] or Natural Questions [8], has an exhaustive list of all possible questions per sentence, except for Monserrate [13], which we use in our evaluation.

Our evaluation is also the first one to focus on all of the following aspects: diversity, correctness and completeness of the generated QA set, the later being traditionally overlooked property of the generative models. Using synthethic data generated with NER allowed

* Corresponding Author. Email: svvakul@amazon.com.

us to evaluate recall of the reference QA set. The results show that our approach is able to strike a fine balance between completeness and diversity, i.e., able to match the reference set without generating duplicate questions. Such cost efficiency is vital for auto-regressive models aiming to minimise the number of redundant token generations.

Our contributions in this paper are three-fold:

- an automated approach for generating a multi-reference corpora of QA pairs using a pre-trained NER model;
- a new public dataset generated using the proposed approach as well as the code required to reproduce and extend it;
- modifications to the standard training and decoding procedures that allow us to train an end-to-end QAG model able to generate a set of diverse QA pairs as confirmed by the results of our comprehensive experimental evaluation[1].

2 Background

There are two types of question generation tasks: answer-aware and answer-agnostic [19]. In the first case, the answer is given defining the focus of the generated question. The latter case is more realistic since the answers are not given in practice and the model should be able to generate both questions and answers.

2.1 Question-answer generation

Given the context C as input (such as a sentence or a paragraph of text), the goal of the QAG model f_{QAG} is to generate a question-answer pair as an output:

$$f_{QAG}(C) =< Q, A > \tag{1}$$

The state-of-the-art approaches to question generation consider this as a sequence-to-sequence task. For our setup, we choose a sequence-to-sequence model that is trained end-to-end to generate the QA pairs jointly, as a single demarcated output sequence. It has furthermore been shown advantageous to generate a QA pair in reverse order by first producing the answer given the context and then producing the question given both the context and the answer embeddings just produced by the model [5]. This approach effectively makes question generation answer-aware and conditioned on the latent answer representation so as to account for variance in prediction and to help train both answer and question generation at the same time.

2.2 Decoding

A standard sequence-to-sequence model uses autoregressive generation. At every step in decoding, the model produces a probability distribution over the entire vocabulary, i.e., the probability for each of the possible tokens to be generated given the input and the output sequence that was already generated so far. For every token at position i, the model produces a distribution such that for the j-th token in the vocabulary $v_j \in V$ there is a probability assigned to its appearance at this position as p_{ij}, where

$$p_{ij} = P(t_i = v_j | t_0 ... t_{i-1}) \tag{2}$$

[1] The annotation guidelines, datasets and code for our experiments are available at: https://github.com/amazon-science/multi-reference-qa-generation.

The simplest decoding approach is called greedy because it always picks the token with the highest probability at each position i using `argmax`. As an alternative, sampling approaches, which make use of the entire probability distribution, have been shown to outperform greedy decoding in practice [2, 6]. To avoid sampling low-probability tokens from the tail of the distribution, the distribution is usually truncated, typically to the top-k tokens as ranked by their probabilities. This decoding approach is called *top-k sampling* and is often applied in practice [2]. The tokens are sampled from the truncated distribution:

$$\begin{gathered} V_i^{topk} = \{v_j \in V : |\{j' : p_{i,j'} > p_{i,j}\}| \leq k\} \\ \widetilde{p}_{i,j} = \frac{p_{i,j}}{\sum_{j' \in V_i^{topk}} p_{i,j'}} \text{ for } j \in V_i^{topk} \\ t_i \sim \widetilde{p}_{i,j} \end{gathered} \tag{3}$$

where $V_i^{topk} \subset V$ are the k most probable tokens predicted by the model for position i, and t_i is the token produced at position i by drawing from V_i^{topk} according to $\widetilde{p}_{i,j}$.

2.3 Training

The model is trained using the cross-entropy (CE) loss that assigns the highest probability to the exact next token as given in the ground-truth reference sequence:

$$CE_O = -\sum_{i=1}^{|O|} \sum_{j=1}^{|V|} t_{ij}^* \log p_{ij} \tag{4}$$

where t_i^* is a $|V|$-dimensional one-hot vector encoding the ground truth target token at position i, and $|O|$ and $|V|$ are the number of tokens in the output sequence and in the vocabulary, respectively.

3 Approach

Our approach consists of the three main parts:

(1) an *augmentation* procedure that generates a dataset with multiple distinct QA reference pairs per input sequence;

(2) a *training* procedure that makes use of those reference pairs by modifying the target output distribution; and

(3) a *decoding* procedure that allows us to produce multiple distinct QA pairs per input sequence using this output distribution.

3.1 Multi-reference augmentation

The first step of our approach is generating a dataset that we use for training and evaluation of the QAG models. We employ a pipeline approach to achieve this by using the output of a separate answer extraction model as input to our pre-trained QAG model.

First, the answer extraction model f_{AE} produces a set of answer spans derived from the input context:

$$f_{AE}(C) = \{A_0, ..., A_l, ..., A_n\}, \text{where } A_l \subset C \tag{5}$$

In our experiments, we use an off-the-shelf NER model as our answer extractor from sentences but it could be another model, such as a summarization model, which identifies salient text spans.

Secondly, for each of the extracted answers we proceed to generate the rest of the output sequence, i.e., the corresponding question:

$$\forall A_l \in f_{AE}(C) : Q_l = f_{QAG}(< C, A_l >) \tag{6}$$

It is possible to reuse the same model trained on the original QAG task defined in (1) for this task as well by simply using each of the answers A_l separately as an input to the decoder of the QAG model. For example, given an input sentence from a news article: *"Comcast rolled out a Web-based on-demand television and movie service on Tuesday that gives customers access to more than 2,000 hours of television and movies."*, we first use NER to detect all named entities in the sentence: "Comcast", "Tuesday" and "more than 2,000 hours". Then, we use each of those named entities as input to the decoder of a pre-trained model to generate the rest of the output sequence as the corresponding question. We generate a single question for each of the answer-entities, resulting in three QA pairs for this example, such as:

f_{QAG} *("Comcast rolled out a Web-based on-demand television and movie service on Tuesday that gives customers access to more than 2,000 hours of television and movies."[BOS]"**a**> Comcast **q**> ") = "Who rolled out a Web-based on-demand television and movie service?"*

Thereby, we obtain a *set* of n QA pairs and recast the original QAG task defined in (1) into the updated QAG task f_{QAG} where the output target is a set of QA pairs instead of a single QA pair:

$$f_{QAG}(C) = \{< Q_l, A_l >\}_{l=[1;n]} \tag{7}$$

For a sequence-to-sequence QAG model we described in Section 2.1, one pair is produced by generating a text sequence starting from the answer and followed by the question with a separator in-between. It means that given a single input sequence C the goal of the model is now to produce a *set* of n output sequences instead:

$$f_{QAG^*}(C) = \{A_l \cdot Q_l\}_{l=[1;n]} \tag{8}$$

where $\cdot$ indicates concatenation.

3.2 Uniform training

Similar to the standard sequence-to-sequence training, we use the target question-answer pairs for training. Our proposed modification accounts for multiple correct output sequences in the case of question generation making use of the positional information of the answer in the beginning of the output sequence.

To train the model on n QA reference pairs, we propose to modify the target output distribution for the first token in the output sequence t_0^*, which is used in the CE loss, see Equation 4 in Section 2.3. More specifically, given a training sample $(C, \{A_l \cdot Q_l\}_{l=[1;n]})$ we construct a subset of tokens:

$$T_A = \cup_{l=1}^{n} \{A_l[0]\} \tag{9}$$

i.e., the set consisting of the first tokens of every answer sequence.

In uniform training, the target distribution in Equation 4 is modified so that in position 0 it becomes

$$t^*_{0,j} = \begin{cases} \frac{1}{|T_A|} & v_j \in T_A \\ 0 & v_j \notin T_A \end{cases} \tag{10}$$

Hence, instead of training the model to produce a single token with probability 1, we are training it to produce *all* tokens in T_A with the uniform probability distribution among them and with 0 probability assigned to the tokens outside of this set (see Figure 1 (a). Thereby, we communicate to the model that all the answers are equally likely to begin at this step.

For the example that was introduced in Section 3.1, the model will be trained to predict the first tokens of the answers: "_Com", "_Tuesday" and "_more" given the sentence as input to the encoder and prefix "a> " as input to the decoder.

We use uniform training only on the set T_A, which is constructed using the first tokens of the output sequences. The rest of the sequences are generated using the standard teacher forcing approach. We implemented this training procedure via multi-task learning, in which different training batches are mixed in equal proportions. Importantly, T_A is used only for training, at inference time the method is agnostic to the answer options and generates a set of QA pairs in parallel.

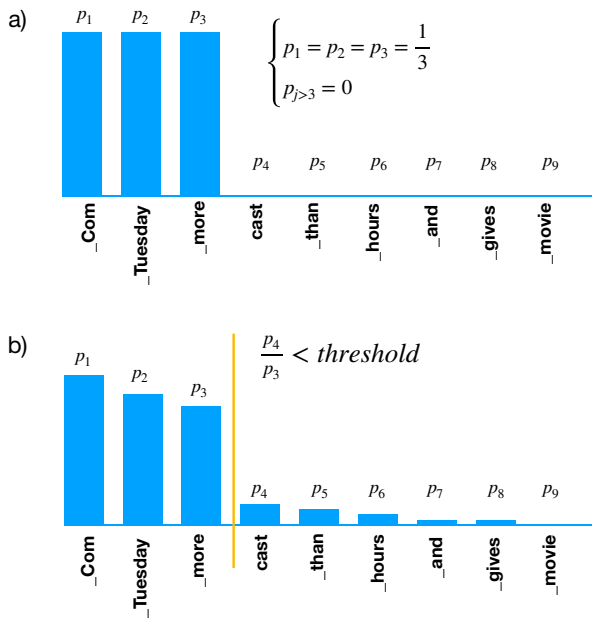

Figure 1: Modified training and decoding procedures for generation of multiple question answer pairs: (a) uniform training, and (b) marginal decoding.

3.3 Marginal decoding

To better leverage the training procedure described in the previous section, we devise a matched decoding procedure that builds on the expectation that, at the first step of decoding, the model will produce a probability distribution with the shape described in Equation 10. More specifically, we are looking for the top n tokens predicted for t_0, the token in the first position of the output sequence:

$$T'_A = V_0^{topk} \tag{11}$$

following Equation 3.

After the set T'_A is produced, we then proceed to generate a set of n sequences independently each starting with a different token from the set: $t_{0,j} \in T'_A$. We can use any of the standard decoding approaches available, such as greedy decoding or beam search, to decode the rest of each sequence.

Since the number of such tokens, i.e., the number of QA pairs, is unknown at inference time, n can be chosen as a hyper-parameter. This approach makes the output similar to other decoding approaches that produce multiple sequences by sampling from the distribution. However, a more sensible approach is to make use of the distribution

P itself to determine the cut-off threshold by looking at the sorted token probabilities themselves (see Figure 1 (b).

The probability threshold could be set as an absolute value, e.g., stop considering answer tokens if the difference between successive probabilities is more than a threshold value. However, since the probabilities assigned to the correct tokens vary in our case, depending on the size of the correct answer set (as specified per Equation 10), it is more appropriate to compare probability ratios instead. A snippet with the pseudo-code for our decoding algorithm with a relative probability threshold is given in Algoritm 1.

Algorithm 1: Marginal decoding

```
Input: C, threshold ;
;
// produce and sort the probability distribution for t_0 ;
P ← f_QAG*(C) ;
P ← sort(P) ;
;
// add the most probable token for t_0 to a set ;
T'_A ← { } ;
(t, lastp) ← P.pop() ;
T'_A.add(t) ;
;
// keep iterating over the sorted probability distribution
// adding the most probable tokens until the threshold is
   reached
for (t, p) in P do
    if p / lastp < threshold then
        break ;
    T'_A.add(t) ;
    lastp ← p ;
;
// produce the rest of the output sequences using a standard
   decoding approach ;
outputs ← { } ;
for t in T'_A do
    s ← C + "[BOS]" + t ;
    outputs.add(decode(f_QAG*, s))
```

The relative threshold changes the number of QA pairs produced by the model based on the output distribution of the first answer token. The more samples we produce the more likely we would fit the ground truth samples, i.e., recall increases but we are also more likely to generate duplicates, i.e., diversity decreases. We experimented with different thresholds but set it in our experiments such that our model produces on average the same number of QA pairs as the baseline approaches to ensure a fair comparison.

4 Automated Evaluation

Our evaluation results are designed to reflect the ability of the QAG model to generate a complete set of question-answer pairs given an input sentence. For every QA pair in the ground truth, we find the most similar QA pair produced by the QAG model.

Note that we had to modify ROUGE calculation for this task to work with the multi-reference dataset. It measures correctness for each of the QA sample from the ground truth separately and hence also indicates recall of the whole set (see Section 4.4 for the description of the metric). We manually examine the generated samples that differ from ground truth in more detail in Section 5.

4.1 Datasets

We use two datasets for evaluation: a large generated dataset, NewsQA-Entities, and a small manually annotated, Monserrate.

NewsQA-Entities is the benchmark we generated using the approach proposed in Section 3.1. The original NewsQA dataset is not a multi-reference dataset. It contains a single question per sentence for the majority of questions. We augment it with synthetic question-answer pairs to simulate a multi-reference dataset that can be then used both for training and evaluation purposes.

For every sentence of the article that contain answers to questions from NewsQA, i.e., potentially newsworthy for the reader, we detect mentioned entities using a pre-trained NER model. Those entity mentions are then used as prefixes (short answers a>) for our QAG model to generate questions.

To produce this benchmark, we use:

- the test and train splits of NewsQA [15], which contains manually produced QA pairs for sentences from news articles;
- Stanza English for NER[2];
- T5 base QAG model fine-tuned on the train split of NewsQA with a greedy decoding strategy.

Stanza was chosen as a state-of-the-art NER model with F1 score of 92% on news articles [11]. In our approach, this model can be substituted with a summarizer, clause extractor or manual human annotations, whichever is deemed more appropriate to provide examples of desired answers for the given dataset.

We select only the sentences that contain at least one entity detected by Stanza and use a random subset of 11K such sentences for training the model and 5k sentences for testing. 41% sentences in the test split contain more than two named entities (see Table 1 for other summary statistics).

Monserrate is a corpus specifically built to evaluate question generation systems [13]. It contains sentences that originate from two English texts about Monserrate palace and sets of questions manually annotated for each of those sentences. The answers to those questions are not provided. The questions in Monserrate are designed to provide an exhaustive reference set for each of the input sentences. The authors use it to evaluate performance of the state-of-the-art systems but do not quantify how many of the reference questions those systems can actually recall.

Monserrate has only 73 sentences annotated and hence is not suitable for training the model. There are, on average, 26 questions per sentence, and 1,933 questions, in total. In contrast with our NewsQA-Entities datasets, the questions in Monserrate tend to overlap, i.e., many of the questions are paraphrases rather than semantically distinct questions. Some of the questions in Monserrate are unanswerable given the reference sentence. Therefore, we prepare an annotation task to filter out such questions and group paraphrases into distinct subsets.

The annotators resolved all the disagreements (Cohen's kappa of 0.39) by discussing the samples that were labeled differently. After the annotation was completed, we obtained a new version of the dataset with 9 semantically distinct questions per sentence, on average, and 647 questions, in total. We use all the questions including paraphrases for matching the generated questions but evaluate recall on the distinct questions only.

[2] https://stanfordnlp.github.io/stanza

Table 1: Statistics of the NewsQA-Entities dataset with the most common entity types. Abbreviations: GPE (Countries/cities/states), ORG (Organization), NORP (Nationalities/religious/political group).

Split	#Sentences	#QA pairs	PERSON	GPE	ORG	DATE	CARDINAL	NORP	TIME	ORDINAL
train	11,203	29,024	27%	17%	15%	14%	8%	6%	2%	2%
test	5,000	13,086	25%	16%	16%	15%	9%	6%	2%	2%

4.2 Models

We evaluate the following three models fine-tuned on the QAG task: 1) the model trained on the original NewsQA dataset; 2) the same model further fine-tuned on our multi-reference dataset using the standard approach; 3) the same model but fine-tuned using the proposed uniform training approach.

Trained on NewsQA. A T5-base model for autoregressive text generation trained on the original NewsQA dataset. We use sentences containing the answers as input and concatenated QA pairs as output of the model. This is also the same model that was used for data augmentation to produce the NewsQA-Entities dataset.

Fine-tuned on NewsQA-Entities. The same baseline model further fine-tuned on the generated NewsQA-Entities dataset.

Fine-tuned on NewsQA-Entities + Uniform. We observe that while fine-tuning the model using uniform distribution (Section 3.2) over the first answer tokens, makes the model forget the original task. Therefore, we train it in the multi-task scenario by mixing both types of batches in equal proportions.

All three models were trained with the maximum input length of 512 tokens and the maximum output length of 1,024 tokens. We selected these sequence lengths by calculating the dataset statistics of the NewsQA training split. The maximum number of epochs was set to two with early stopping if the loss converges before that.

4.3 Decoding strategies

We test the standard decoding strategies against our approach.

Greedy: always picks the argmax token and generates a single output sequence.

Beam search: top-k hypotheses with max product probability (we set number of beams to 5 in our experiments).

Top-k sampling: samples from the top-k tokens ordered by probability (we set k=50 in our experiments).

Top-p (nucleus) **sampling**: considers the top tokens for which the sum of their probabilities is at least p (in our experiments p=0.92).

Marginal decoding: our approach described in Section 3.3.

For marginal decoding we generate the same number of QA pairs as the maximum number of QA pairs per sentence based on the dataset statistics (7 for NewsQA-Entities and 21 for Monserrate) since it uses a cut-off threshold to determine the number of output sequences dynamically. For all other sampling methods, except for greedy search, we use the same number of output sequences as generated by the marginal decoding approach to make their results more comparable. Since the number of QA samples have a direct effect on the performance metrics, optimising for it is important for the end task but for the purpose of our evaluation it should be fixed.

4.4 Evaluation metrics

We are evaluating diversity of the generated questions and answers separately and also assessing how many of the ground-truth sample our generative model can recall.

Distinct-1 is the standard metric used for measuring diversity in NLG, e.g. for the dialogue response generation task [9]. It is defined as the number of distinct tokens (unigrams) divided by the total number of generated tokens. We calculate it by measuring the proportion of the unique tokens across all the sequences generated for the same input sequence. Greedy decoding always produces a single output sequence. Hence, its diversity metric is always 1.

ROUGE is the standard metric used in summarization. We calculate ROUGE-L F-measure that measures the token overlap for the longest common subsequence of a pair of strings. Since we have multiple different reference sequences, we take the maximum ROUGE score calculated across all the predicted sequences. Then, we compute the mean of these ROUGE scores across all the test samples. Note that we measure ROUGE with respect to each of the reference sequences to evaluate how well we can recall the ground-truth questions and answers on average. Hence, the number of scores being aggregated will always be the same irrespective of the number of questions generated by the model. We measure ROUGE only for questions with distinct answers to avoid matching the same question paraphrases multiple times.

softM0.5 shows a proportion of the generated sequences that have a certain overlap with the target sequence, which is defined by the minimum ROUGE score. We use ROUGE-L F-measure of 0.5 here to count the sequence as a partial match for both questions and answers. This is a novel metric we introduce to provide more insight into our results. Since ROUGE scores are aggregated across all references, it is impossible to tell whether some of the references were matched very well and others not at all, or all of the references were matched to some extent. Thereby this soft match score allows to estimate the percentage of references that we consider as a partial match determined by the similarity threshold.

SBERT measures semantic similarity between questions beyond the token overlap measured by ROUGE. We use a pre-trained model available from the SentenceTransformer library trained on the Quora Duplicate Questions dataset using DistilBERT base model.[3].

4.5 Results on NewsQA-Entities

Results of the automated evaluation are given in Table 2. The table demonstrates that both questions and answers generated using our approach are on average more diverse and match the ground truth better than all the other baselines (see last raw of this table).

Most of the answers produced by the original model trained on NewsQA are not entities. Often those are rather longer spans extracted from the input sentence (see Table 4).

Greedy decoding produces only a single output sequence and, therefore, receives the diversity score of 1 for answers and questions. While both top-p and top-k sampling lead to lexically diverse questions, their answers tend to overlap.

Marginal decoding in the combination with uniform training (last row of Table 2) has the highest diversity scores and fits the ground-truth data better (ROUGE-L mean), especially in terms of the answer spans. This means that our approach succeeds in emulating the NER-based generation pipeline and produces diverse questions with entities as their answers.

[3] https://www.sbert.net

Table 2: Evaluation results on NewsQA-Entities. #QA is the average number of QA pairs generated per input sentence.

Model	Decoding	#QA	Answers			Questions			
			Distinct-1	ROUGE	softM0.5	Distinct-1	ROUGE	softM0.5	SBERT
Ground truth		2.6	**0.99**	**1**	**1**	**0.78**	**1**	**1**	**1**
Trained on NewsQA	Greedy	**1**	**1**	0.25	0.24	1	0.43	0.33	0.61
	Beam search	3	0.42	0.25	0.19	0.51	0.37	0.27	0.58
	Top-k sampling	3	0.61	0.37	0.33	**0.75**	0.38	0.26	0.62
	Top-p sampling	3	0.6	0.36	0.33	0.74	0.38	0.25	0.61
	Marginal decoding	2.7	0.73	0.36	0.31	0.67	0.43	0.34	0.62
Fine-tuned on NewsQA-Entities	Greedy	**1**	**1**	0.38	0.40	**1**	0.50	0.44	0.66
	Beam search	3	0.57	0.62	0.63	0.54	**0.67**	0.64	**0.78**
	Top-k sampling	3	0.58	0.57	0.57	0.55	0.61	0.59	0.75
	Top-p sampling	3	0.56	0.58	0.58	0.53	0.62	0.59	0.75
	Marginal decoding	2.4	**0.93**	0.66	0.66	0.74	0.60	0.57	0.72
Fine-tuned on NewsQA-Entities + Uniform	Greedy	**1**	**1**	0.38	0.38	**1**	0.50	0.43	0.66
	Beam search	3	0.57	0.64	0.64	0.54	**0.67**	0.65	**0.78**
	Top-k sampling	3	0.57	0.59	0.59	0.56	0.61	0.58	0.75
	Top-p sampling	3	0.55	0.59	0.59	0.53	0.62	0.60	0.75
	Marginal decoding	2.8	**0.93**	**0.90**	**0.90**	0.71	**0.67**	**0.68**	**0.78**

While questions decoded with beam search also have a high overlap with the ground truth, they are less diverse than the ones produced by our approach. Closer examination of the results confirms that the QA pairs generated with beam search are paraphrases.

4.6 Results on Monserrate

In addition, we evaluate our approach on the manually produced benchmark dataset (see Table 3). Since Monserrate does not have answers annotated, we evaluated only diversity of the produced answers and measure both diversity and overlap with the ground truth for the generated questions.

The first thing to note is that our approach generates fewer QA pairs than were produced by human annotators. This is an expected result since not all of the questions in Monserrate are entity-based, however we still manage to match approximately a third of those questions. Marginal decoding again shows the best results in terms of producing diverse QA pairs that match the ground-truth data.

We manually inspected the generated QA pairs and concluded that all the entity-answers present in the dataset were identified correctly. The ground truth also contains the questions that address those entity-answers. Some of the generated questions, however, do not match the answers or the input sentence well. We dedicate the next section to a more systematic analysis that quantifies different types of errors our model makes.

5 Error Analysis

To further strengthen the results of our automated evaluation, we conduct manual annotation of the QA pairs produced by our approach. Since the automated metrics indicate the discrepancy between the produced sequences and the ground-truth sets, we use them to identify cases that deviate from the ground truth (with ROUGE score less than 0.5) and examine them in more detail.

The analysis was performed on a sample of the NewsQA-Entities dataset. We sample a set of answers that deviate from the ground truth using our softM0.5 metric and questions that have the same answers but differ according to softM0.5. Using a small subset of the results we first identify the most common error types and then employ a third-party team of professional annotators to perform an independent analysis of the produced results. We obtained 238 annotated answers and 201 annotated questions, in total. Results indicate that the majority of questions and answers are correct even when they deviate from the ground truth.

The majority of answers are either entities (44%) or entities with context spans surrounding those entities (13%). 43% of the answers are not entities but spans extracted from the input sentence. This result suggests that our model learned to identify new entities and potential answers that play a similar role in the sentence but cannot be detected by the original NER model. The most common mistake the model makes when generating the answers is by truncating them too soon (24%). 16% of the answers produced were not found in the input text and constitute proper mistake which is a known drawback of generative models.

The errors identified at the question generation phase are mostly the questions that do not have an answer in the input sentence (21%) or do not match the answer produced by the model (9%). These are the same error types, which the original model suffers from as well in similar proportions.

Table 4 provides examples of the generated QA samples alongside with the input sentences. It demonstrates that the baseline approaches are more prone to producing duplicates and question paraphrases than our approach.

6 Related Work

A recent survey on question generation provides a comprehensive overview of the research work alongside multiple dimensions [19]. It clearly points out that improving diversity of the generated questions as well as training with multi-reference data are bot important directions that are largely overlooked by the current approaches. This is the exact research gap which we are addressing with this work.

While multi-reference training for QAG was not previously studied in the related work, it has been already applied in other natural language generation tasks, such as machine translation and dialogue response generation. In [20], the authors generated pseudo-references for machine translation and image captioning task and then convert this multiple reference dataset into a single reference dataset for training the model. We consider this approach as our baseline (Fine-tuned on NewsQA-Entities).

To bootstrap a multi-reference dataset for training and evaluation of our model, we make use of a pre-trained NER model and many previously proposed QAG approaches utilised named entities. In [9] question templates are filled with entities from the input text when

Table 3: Evaluation results of our model fine-tuned on NewsQA-Entities with uniform training on the Monserrate corpus. #QA refers to the average number of QA pairs generated per input sentence.

Decoding	#QA pairs	Answers	Questions		
		Distinct-1	Distinct-1	ROUGE-L mean	softM0.5
Ground truth	9	-	0.47	**1**	**1**
Greedy	1	**1**	**1**	0.35	0.19
Beam search	4	0.51	0.44	0.42	0.30
Top-k sampling	4	0.52	0.46	0.42	0.30
Top-p sampling	4	0.49	0.45	0.41	0.27
Marginal decoding	4	**0.83**	**0.56**	**0.45**	**0.33**

Table 4: Examples of the generated QA pairs. Duplicate questions and answers are highlighted **in bold**.

Input sentence	Ground truth	Trained on NewsQA	Fine-tuned on NewsQA-entities with beam search	Fine-tuned on NewsQA-entities + uniform with marginal decoding
The service, named Fancast XFINITY TV (formerly called TV Everywhere), is the biggest cable industry initiative to keep people from skipping traditional TV service in the United States.	a> TV Everywhere) q> **What is the name of the new service?** a> Fancast XFINITY TV q> **What is the name of the new service?** a> the United States q> Where is the TV service?	a> **To keep people from skipping traditional TV service in the United States.** q> What did the program do? a> XFINITY TV q> What does the company want people to skip? a> **to keep people from skipping traditional TV service in the United States.** q> what is the fancast xfinity television?	a> Fancast XFINITY TV q> What is the name of the service? a> **the United States** q> Where is Fancast XFINITY TV? a> **the United States** q> Where is the service located?	a> Fancast XFINITY TV q> What is the name of the service? a> the United States q> Where is the TV service?
And also my time is important, so sitting down and watching a show at 8 or 9 p.m.	a> 8 or 9 p.m q> When is the show?	a> **8 or 9 p.m.** q> Time it's important what for him? a> And also my time is important, so sitting down and watching q> What did the girl says? a> **8 or 9 p.m.** q> When is that time a priority?	a> 8 or 9 p.m q> **When is the show?** a> 8 q> What time is important? a> 9 p.m q> **When is the show?**	a> 8 or 9 p.m q> When is the show?
Networks have tried for the past couple of years to find a way to reach Web-watching audiences by streaming content on their Web sites or making partnerships with Hulu, one of the larger online TV sites.	a> Hulu q> What is the name of the TV site? a> the past couple of years q> How long have networks tried to reach Web-watching audiences?	a> Hulu, one of the larger online TV sites. q> What companies are trying to do what with their content? a> to find a way to reach Web-watching audiences by streaming content on their Web sites or making partnerships with Hulu, q> What has the Nets tried? a> Networks q> Who has tried for years to reach Internet watchers?	a> **the past couple of years** q> **When did Networks try to reach Web-watching audiences?** a> **the past couple of years** q> How long have networks tried to reach Web-watching audiences? a> **the past couple of years** q> **When did the networks try to reach Web-watching audiences?**	a> Hulu q> What is one of the larger online TV sites? a> **the past couple of years** q> **When did Networks try to find a way to reach Web-watching audiences?** a> **The past couple of years** q> **When did Networks try to reach Web-watching audiences?** a> Networks q> Who has tried to reach Web-watching audiences? a> Internet q> What kind of audiences do networks try to reach? a> one q> What is Hulu?

designing an automatic tutoring systems. None of those approaches, however, considered training a QAG model end-to-end using entities as a multi-reference QA dataset.

7 Conclusion

We proposed a new approach for generating diverse question-answer pairs and validated it in an experimental evaluation. Our end-to-end QAG model dynamically identifies question-worthy context words pre-trained on named entities and uses them to condition subsequent question generation. Our results suggest that the proposed approach is successful in generating the same answer-entities as in the ground truth produced by a pre-trained NER model as well as in identifying new question-worthy phrases beyond named entities. A manual examination confirms that the generated questions are correct even when they differ from the ground-truth generated by the baseline.

We showed that it is possible to train a single model using data generated by a pipeline approach and replace this pipeline approach to reduce friction. In our case, the NER-based heuristic was used to demonstrate our proposed approach allowing us to avoid manual collection of a new multi-reference dataset. It is also straight-forward to extend our NER-based heuristic with other answer extractors or generators. However, our model can also be trained on human labeled data to produce QA pairs that cannot be generated by the pipeline approach. For example, a human-in-the-loop approach allows to use NER and other similar heuristics to bootstrap data collection but modify generated samples by selecting only useful questions.

Acknowledgments

We thank Gianni Barlacchi for sharing with us the pre-processed version of the NewsQA dataset with answers mapped to the original sentences from the news articles.

References

[1] Nan Duan, Duyu Tang, Peng Chen, and Ming Zhou, 'Question generation for question answering', in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pp. 866–874. Association for Computational Linguistics, (2017).

[2] Angela Fan, Mike Lewis, and Yann N. Dauphin, 'Hierarchical neural story generation', in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pp. 889–898. Association for Computational Linguistics, (2018).

[3] Yuwei Fang, Shuohang Wang, Zhe Gan, Siqi Sun, and Jingjing Liu, 'Accelerating real-time question answering via question generation', *CoRR*, **abs/2009.05167**, (2020).

[4] Zichu Fei, Qi Zhang, Tao Gui, Di Liang, Sirui Wang, Wei Wu, and Xuanjing Huang, 'CQG: A simple and effective controlled generation framework for multi-hop question generation', in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pp. 6896–6906. Association for Computational Linguistics, (2022).

[5] Jing Gu, Mostafa Mirshekari, Zhou Yu, and Aaron Sisto, 'Chaincqg: Flow-aware conversational question generation', in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pp. 2061–2070. Association for Computational Linguistics, (2021).

[6] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi, 'The curious case of neural text degeneration', in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, (2020).

[7] Robin Jia, Mike Lewis, and Luke Zettlemoyer, 'Question answering infused pre-training of general-purpose contextualized representations', in *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, eds., Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, pp. 711–728. Association for Computational Linguistics, (2022).

[8] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov, 'Natural questions: a benchmark for question answering research', *Transactions of the Association of Computational Linguistics*, (2019).

[9] David Lindberg, Fred Popowich, John C. Nesbit, and Philip H. Winne, 'Generating natural language questions to support learning on-line', in *ENLG 2013 - Proceedings of the 14th European Workshop on Natural Language Generation, August 8-9, 2013, Sofia, Bulgaria*, pp. 105–114. The Association for Computer Linguistics, (2013).

[10] Rodrigo Frassetto Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho, 'Document expansion by query prediction', *CoRR*, **abs/1904.08375**, (2019).

[11] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning, 'Stanza: A Python natural language processing toolkit for many human languages', in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, (2020).

[12] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang, 'SQuAD: 100,000+ questions for machine comprehension of text', in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, (2016).

[13] Hugo Rodrigues, Eric Nyberg, and Luísa Coheur, 'Towards the benchmarking of question generation: introducing the monserrate corpus', *Lang. Resour. Evaluation*, **56**(2), 573–591, (2022).

[14] Xiaoyu Shen, Gianni Barlacchi, Marco Del Tredici, Weiwei Cheng, Bill Byrne, and Adrià Gispert, 'Product answer generation from heterogeneous sources: A new benchmark and best practices', in *Proceedings of The Fifth Workshop on e-Commerce and NLP (ECNLP 5)*, pp. 99–110, Dublin, Ireland, (May 2022). Association for Computational Linguistics.

[15] Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman, 'Newsqa: A machine comprehension dataset', in *Proceedings of the 2nd Workshop on Representation Learning for NLP, Rep4NLP@ACL 2017, Vancouver, Canada, August 3, 2017*, pp. 191–200. Association for Computational Linguistics, (2017).

[16] Huazheng Wang, Zhe Gan, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, and Hongning Wang, 'Adversarial domain adaptation for machine reading comprehension', 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 2019, Hong Kong, China, November 3-7, 2019*, eds., Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, pp. 2510–2520. Association for Computational Linguistics, (2019).

[17] Kexin Wang, Nandan Thakur, Nils Reimers, and Iryna Gurevych, 'GPL: generative pseudo labeling for unsupervised domain adaptation of dense retrieval', in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pp. 2345–2360. Association for Computational Linguistics, (2022).

[18] Bingsheng Yao, Dakuo Wang, Tongshuang Wu, Zheng Zhang, Toby Jia-Jun Li, Mo Yu, and Ying Xu, '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), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pp. 731–744. Association for Computational Linguistics, (2022).

[19] Ruqing Zhang, Jiafeng Guo, Lu Chen, Yixing Fan, and Xueqi Cheng, 'A review on question generation from natural language text', *ACM Trans. Inf. Syst.*, **40**(1), 14:1–14:43, (2022).

[20] Renjie Zheng, Mingbo Ma, and Liang Huang, 'Multi-reference training with pseudo-references for neural translation and text generation', in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pp. 3188–3197. Association for Computational Linguistics, (2018).