On Generating Fact-Infused Question Variations

Arthur Deschamps*
ByteDance Ltd.
Singapore
arthur.deschamps1208@gmail.com

Sujatha Das Gollapalli*, See-Kiong Ng
Institute of Data Science
National University of Singapore
{idssdg,seekiong}@nus.edu.sg

Abstract

To fully model human-like ability to ask questions, automatic question generation (QG) models must be able to produce multiple expressions of the same question with different levels of detail. Unfortunately, existing datasets available for learning QG do not include paraphrases or question variations affecting a model's ability to learn this capability. We present *FIRS*, a dataset containing human-generated **f**act-**i**nfused **r**ewrites of questions from the widely-used **S**QuAD dataset to address this limitation. Questions in *FIRS* were obtained by combining a given question with facts of entities referenced in the question. We study a double encoder-decoder model, **F**act-**I**nfused **Q**uestion **G**enerator (*FIQG*), for learning to generate fact-infused questions from a given question. Experimental results show that *FIQG* effectively incorporates information from facts to add more detail to a given question. To the best of our knowledge, ours is the first study to present fact-infusion as a novel form of question paraphrasing.

1 Introduction

Recently, automatic Question Generation (QG) is being addressed for generating natural language questions for a given input text passage. Viewed as the reverse of the well-studied question answering task (QA), QG has been applied in education and tutoring (Heilman and Smith, 2010; Lindberg et al., 2013), dialog systems and chatbots (Shum et al., 2018), as well as for improving QA systems (Duan et al., 2017; Tang et al., 2018).

Various deep learning models are being rapidly developed for QG (Talmor and Berant, 2018; Kim et al., 2019; Tuan et al., 2020; Pan et al., 2020; Su et al., 2020; Wang et al., 2020a). However, it is only recently that QG studies have started focusing on an important aspect of the human question generation process known as *paraphrasing*, or the ability to ask questions in diverse ways all expressing the same intent (Harrison and Walker, 2018; Wang et al., 2020b).

Paraphrasing ability has been identified as a necessary aspect of learning human-like language generation (Shum et al., 2018; Huang et al., 2020) and was previously studied in context of community QA (Liang et al., 2016; Kunneman et al., 2019; Hosking and Lapata, 2021). These works address the identification of synonymous and syntactic question variations such as ("What's the weight of an elephant in kg?"; "How heavy is an elephant?"). In addition to synonymous variations, human beings are also adept at generating questions expressing the same intent with varying level of details. For example, consider a QA pair from the SQuAD dataset[1] shown in Table 1. SQuAD is one of the widely-used datasets for training QG models and contains about 100K training instances made up of an answer context, the answer string, and a "correct" question (Rajpurkar et al., 2016). We show in Table 1, fact-infused rewrites (or alternatively, variations) for the SQuAD question: "In what year did IBM get its name?". Except question 2 which is a synonymous variation, the other variations include additional details of the entity "IBM" obtained from Google's Entity Search API.[2]

We argue that question variations that include more detail can provide a form of query expansion and are likely to benefit downstream applications. Indeed, it has been observed that content words and named-entities referenced in the question improve the answerability of a question (Nema and Khapra, 2018) and result in improved QA and reading comprehension performance through the addition of

*Equal contribution. All work was done at Institute of Data Science, NUS.

[1] https://rajpurkar.github.io/SQuAD-explorer/

[2] https://developers.google.com/knowledge-graph

Proceedings of Recent Advances in Natural Language Processing, pages 335–345
Sep 1–3, 2021.
https://doi.org/10.26615/978-954-452-072-4_039

SQuAD QA Pair
Passage Title: IBM
Answer Context: The company originated in 1911 as the Computing-Tabulating-Recording Company (CTR) through the consolidation of The Tabulating Machine Company, the International Time Recording Company, the Computing Scale Company and the Bundy Manufacturing Company. CTR was renamed "International Business Machines" in 1924, a name which Thomas J. Watson first used for a CTR Canadian subsidiary. The initialism IBM followed. Securities analysts nicknamed the company Big Blue for its size and common use of the color in products, packaging and its logo.

Question: In what year did IBM get its name?

Google Entity Search Result for the query "IBM": International Business Machines Corporation is an American multinational technology company headquartered in Armonk, New York, with operations in over 170 countries.
Human-generated Fact-Infused Variations:
1. In what year did International Business Machines Corporation get its name?
2. `When` did the IBM get its name?
3. In what year did multinational technology company IBM get its name?
4. In what year did American company IBM get its name?

Table 1: Example from the *FIRS* dataset.

constraints on the candidate answers (Chakrabarti, 2020; Wang et al., 2016; Steuer et al., 2020; Huang et al., 2019). We refer to question variations obtained by incorporating facts of relevant entities into a question as "fact-infused question rewrites". Fact-infused question rewrites add details to an existing question without changing its underlying intent thereby comprising a form of *paraphrasing*.

How can we combine a given question with facts of relevant entities to generate question variations with more details? We address this precise question in our paper and make the following contributions:

1. We present *FIRS*, a novel dataset containing **F**act-**I**nfused **R**ewrites of **S**QuAD questions. *FIRS* contains approximately 6.9K paraphrases of about 1.5K questions that were manually-generated by crowdworkers on the Amazon Mechanical Turk platform.[3]

   Unlike mere synonymous paraphrases available in existing paraphrase datasets such as Quora Question Pairs[4] and WikiAnswers (Fader et al., 2013) or prominent QG datasets such as SQuAD (Rajpurkar et al., 2016), HotPotQA (Yang et al., 2018), ComplexWebQA (Talmor and Berant, 2018), MS MARCO (Nguyen et al., 2016) that only incorporate one reference question for a given passage and answer-span pair, *FIRS* contains multiple question variations obtained by augmenting a given question with different facts of entities referenced in the question. To the best of our knowledge, *FIRS* is a first-of-its-kind dataset available for learning fact-infusion into a given question.[5]

2. We propose **F**act-**I**nfused **Q**uestion **G**enerator (*FIQG*), a novel attention-based sequence-to-sequence model using a double encoder-decoder set-up and an extended copy mechanism for learning to generate fact-infused question rewrites. The performance of *FIQG* is demonstrated on *FIRS* and compared against state-of-the-art QG models modified for fact-infused question rewriting. Our experimental results show that *FIQG* significantly outperforms other models on standard evaluation metrics. *FIRS* and the novel task of fact-infusion not only complement on-going studies on question generation and paraphrasing but also presents new challenges for learning models and evaluation metrics. In addition, we expect *FIRS* to be useful in studying other QA-related tasks due its links with the widely-used SQuAD dataset.

Organization: We summarize our dataset collection process in Section 3. Next, we present *FIQG*, our model for learning to generate fact-infused question variations in Section 4. Section 5 contains a discussion on our experimental settings, results and observations while Section 2 briefly summarizes closely-related works. Finally, we provide future directions with conclusions in Section 6.

[3] https://www.mturk.com/

[4] https://www.kaggle.com/c/quora-question-pairs

[5] *FIRS* and the code used in this paper along with the Appendix can be downloaded for academic use from `https://github.com/NUS-IDS/ranlp21-fiqv`.

2 Related Work

Models for question generation and question answering are being rapidly developed in current NLP research. We refer our reader to a survey by Pan et al. (2019) for an overview on challenges, existing approaches, as well as evaluation metrics for QG. Several QG models use LSTM-based encoder-decoder setups with attention and copy mechanisms (Zhou et al., 2018; Duan et al., 2017; Zhao et al., 2018; Kim et al., 2019). Recent works are focused on improving QG performance by incorporating external knowledge, semantic information, and reinforcement learning into this basic architecture (Nema et al., 2019; Pan et al., 2020; Wang et al., 2020a; Majumder et al., 2021). Other state-of-the-art QG frameworks include variational autoencoders, graph convolutional networks and transformers (Lee et al., 2020; Su et al., 2020; Kriangchaivech and Wangperawong, 2019).

Paraphrase generation is a related task for identifying semantically similar texts in applications such as retrieval and question answering, query reformulation and dialog system applications (Liang et al., 2016; Zhao and Wang, 2010). Similar to QG, seq2seq models and encoder-decoder architectures are common in paraphrase generation works (Gupta et al., 2018) but other approaches for paraphrase generation include variational autoencoders and translation models (Wang et al., 2019; Li et al., 2018; Hosking and Lapata, 2021).

3 *FIRS* Dataset Creation

As highlighted in Section 1, existing datasets for QA/QG and paraphrase generation do not include question variations with details. To fill this gap, we collected a new dataset by integrating the questions in the widely-used Stanford Question Answering Dataset (SQuAD) with relevant facts obtained from Google's Entity Search API as follows:

Collecting candidate question-entity pairs: We selected from SQuAD, questions that refer to named entities in either the (1) question or (2) answer texts. In Table 1, we showed an example where the relevant entity *IBM* is mentioned in the question text. For the second case, consider a question from the SQuAD dataset from a passage on *Computer Security*, namely, "What is the source of the quote?" with the corresponding answer string "Reuters". Nowhere in the SQuAD answer passage for this question is a mention of what "Reuters" is but using its entity description from Google, example paraphrases created by our crowdworkers for this question include "What news agency is the source of the quote?" and "Which international news organization is the source of the quote?". This example highlights how a *vague* "What is" question can be expanded through the addition of the answer type ("news agency") detail.

We obtained the subset of $25,316$ questions from the 100K questions in SQuAD which reference 'tangible' named-entity types such as people, places, and organizations. Entity types referring to concepts such as "quantity, percent etc" are not supported in currently-available knowledge resources. For example, for the *IBM* question in Table 1, it is difficult to obtain focused knowledge pertaining to the answer "1924" (of type "date").

Entity Name: IBM
Type: 'Corporation', 'Thing', 'Organization'
Description: Computer hardware company
Detailed description: International Business Machines Corporation is an American multinational technology company headquartered in Armonk, New York, with operations in over 170 countries.

Table 2: Search Result for "IBM" on the Entity Search API

Obtaining Entity Descriptions: We performed entity searches through the **G**oogle Knowledge Graph **E**ntity **S**earch API (GES) using the entity names as query strings. Next, entity-type match rules and text similarity thresholds were applied based on the source SQuAD passage to identify the correct entity description from the search results. We were able to obtain descriptions for $62,473$ entities referenced in SQuAD questions using the above process. Based on crowdsourced annotations (described next), the precision of our search and filtering is $\sim 97\%$. An example search result from GES along with its different fields is shown for the query "IBM" in Table 2.

We note that compared to other resources such as DBpedia (Lehmann et al., 2015) and YAGO (Hoffart et al., 2013), the coverage of entities and facts is several scales higher in GES.[6] After manually examining hundreds of results, we found GES to be consistently superior and accurate for our purpose. The "detailed description" fields were used by our crowdworkers while creating the question paraphrases. A limitation however is that there is no official documentation on the resources and algorithms employed in GES and neither is the full-

[6] https://en.wikipedia.org/wiki/Knowledge_Graph

type hierarchy information directly available given its proprietary nature.[7]

Creating ground truth paraphrases: We randomly sampled a subset of about 1600 (question, entity) pairs collected from Steps 1 and 2 for obtaining human-generated question variations. We set up our task through the crowdsourcing platform Amazon Mechanical Turk (AMT) following similar dataset collection efforts (Rajpurkar et al., 2016; Yang et al., 2018; Harrison and Walker, 2018). Each question, along with the entity descriptions was examined by three crowdworkers. The answer passage with the answer highlighted was also provided for the workers to identify cases where the entity is not relevant.

We required the crowdworkers to have greater than 95% HIT approval rate, a minimum of 10,000 HITs, and be located in the United States. The workers were instructed to "Rewrite the original question in more details using information from the provided knowledge" and to "Ensure that the intent of the original question remains the same." Several examples of good and bad rewrites along with detailed explanations were included as guidance. At least one and up to three different re-writings were collected for each question per crowdworker.

Split	#Questions	#Rewrites	#Avg
Train	1156	4973	4.30
Dev	128	531	4.14
Test	299	1400	4.63
Total #Questions: 1583, #Paraphrases: 6904			

Table 3: Dataset Summary

	Intra-Set	**w/ Base Question**
SBAK	0.6285±0.2205	0.7412±0.2061
Jaccard	0.4033±0.1448	0.5178±0.1429
POS Tag Spread of Added Words		
Nouns+Proper nouns		37.02%
Adpos+Adj+Det		32.68%
Verbs+Adverbs		9.2%
Other POS		21.1%

Table 4: Properties of *FIRS*

After pooling the results of the AMT task, filtering out duplicates and variations that do not include any word from the extra knowledge (such as rewrite#2 for *IBM* in Table 1), our dataset has an average of four fact-infused variations for each question and is summarized in Table 3. We refer to our dataset as *FIRS* for **F**act-**I**nfused **R**ewrites of **S**QuAD questions.

3.1 Analysis of *FIRS*

We analyzed the question rewrites in *FIRS* along two dimensions, namely, (i) Diversity and (ii) Details. That is, a fact-infused rewrite should retain the semantics of the *base question* (original question from SQuAD) in terms of its intent but have other words that add extra details of relevant entities. To characterize this aspect, we employ the Simple Approximate Bigram Kernel (SBAK) similarity to measure the pairwise similarity between two sentences. Dependency-tree based similarity measures that account for partial matches and type of dependency edges are known to better represent semantic similarity between two sentences compared to bag-of-words similarity functions (Ambati, 2008; Özateş et al., 2016).

In Table 4, the average values of pairwise similarities of the fact-infused question variations with each other are shown in the "Intra-Set" column and with the base question are shown in the third column. The high SBAK similarity is indicative of semantic or intent similarity between the base question and the variations. However, the Jaccard overlap scores between the word sets (computed without stopwords) is lower due to the additional words present in the rewrites.[8]

The percentage spread of the parts-of-speech tags for the words added in rewritten questions are shown in Table 4. Not surprisingly, about 37% of the newly-added words are proper nouns or nouns whereas about 33% of words refer to adpositions, adjectives, and determiners that are often assigned to words surrounding noun phrases.[9] These assignments indicate that the extra words added in rewrites are often content words and therefore, can be expected to improve the answerability of questions (Nema and Khapra, 2018).

Additional Notes on Data Collection: We performed the following checks to meet the *ethics, quality, and reliability* considerations for our collected questions. As part of the AMT data collection process, the *anonymity* and *privacy* of the crowdworkers is already ensured. Furthermore, the settings for the HIT approval rates, and location of

[7] Further details of the search result filtering processes are provided in the Appendix (See Footnote 5).

[8] The formulae from (Özateş et al., 2016) are included in the Appendix for reference.

[9] https://universaldependencies.org

the worker, described previously are set similar to previous QA/QG data collection efforts to ensure the English language skills of the data annotators. A remuneration of \$0.20 per assignment was paid to each worker. A total of 75 workers helped in creating our dataset, with about 47% of the workers labeling less than 5 questions each.

We ensure *quality* of the collected question variations, by employing a set of rules based on the a) similarity with the original question, (b) similarity within the collected paraphrases, (c) presence of the answer token, and (d) length of the rewritten question versus the original question. About 3% of the collected data was filtered out with the above rules. Finally, at every step of data collection, we performed checks manually on random subsets of the collected data to ensure the *reliability* of the named-entity taggers, the entity descriptions filtered from GES results as well as the quality of the rewrites produced by the workers.

4 Generating Fact-Infused Questions

Definition: Given two input sequences of $|n|$ base question words, $Q^b = q_1, q_2 \dots q_n$, and $|m|$ words of a fact related to an entity, $F = f_1, f_2 \dots f_m$, the objective of fact-infused question generation is to generate an output sequence of $|k|$ words, $Q^p = p_1, p_2 \dots p_k$, such that Q^p is a fact-infused rewrite of Q^b. That is, Q^p includes specific details from F while maintaining the intent of Q^b and our goal is to find Q^p that maximizes the conditional likelihood:

$$Q^p = \underset{Q}{argmax}\ \mathcal{P}(Q|Q^b, F)$$

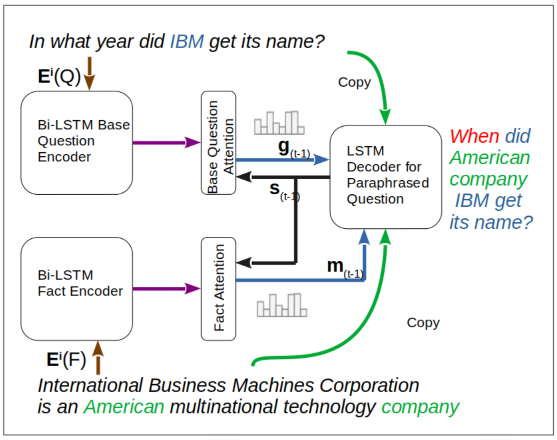
In what year did IBM get its name?
E^i(Q)
Copy
Bi-LSTM Base Question Encoder
Base Question Attention
g(t-1)
s(t-1)
LSTM Decoder for Paraphrased Question
When did American company IBM get its name?
Bi-LSTM Fact Encoder
Fact Attention
m(t-1)
Copy
E^i(F)
International Business Machines Corporation is an American multinational technology company

Figure 1: Fact-Infused Question Generator Network.

We follow standard question generation approaches and adopt an attention-based encoder-decoder architecture for estimating the probability function, $\mathcal{P}$ (Bahdanau et al., 2015; Sutskever et al., 2014; Kim et al., 2019). The main components of our **F**act-**I**nfused **Q**uestion **G**enerator Network (*FIQG*) are depicted in the schematic diagram in Figure 1 and summarized below:

Fact and Question Encoders: We use separate encoders for representing the question and fact sequences. The encoders are one-layer bidirectional LSTMs that extract contextual features from the input question (or alternatively, fact) and represent them as hidden states of the forward and backward LSTMs. Let $\mathcal{E}^i(Q^b)$ and $\mathcal{E}^i(F)$ represent the feature-rich embeddings of our base question and fact, respectively (Zhou et al., 2018). Then,

$$\mathcal{E}^i(Q^b) = [\mathcal{E}^i(q_1), \dots, \mathcal{E}^i(q_n)]$$
$$\mathcal{E}^i(F) = [\mathcal{E}^i(f_1), \dots, \mathcal{E}^i(f_m)]$$

where $\mathcal{E}^i(w)$ refers to the input feature embedding for word w. Using LSTM notations, the hidden state for the fact encoder is therefore given by

$$\mathbf{o}_t^F = [\overrightarrow{h}_t^F; \overleftarrow{h}_t^F]$$

where $\overrightarrow{h}_t^F$ and $\overleftarrow{h}_t^F$ are the hidden vectors of the forward and backward LSTMs, respectively, at time t and ; represents the concatenation operator. The hidden state for the question encoder can be similarly represented as

$$\mathbf{o}_t^{Q^b} = [\overrightarrow{h}_t^{Q^b}; \overleftarrow{h}_t^{Q^b}]$$

Next, applying the attention mechanism (Bahdanau et al., 2015) for the question encoders over its hidden states, the attention weighted sum of the contextualized question can be written as

$$\mathbf{g}_t = \sum_{i=1}^{n} \alpha_{ti} \mathbf{o}_i^{Q^b}$$
$$\alpha_{ti} = \frac{\exp(a_{ti})}{\sum_{k=1}^{n} \exp(a_{tk})}$$
$$a_{ti} = f(\mathbf{s}_{t-1}, \mathbf{o}_i^{Q^b})$$

where α_{ti}s represent the attention weights with parameters a_{ti} such that $\mathbf{Y}_t^{Q^b} = \{\alpha_{ti}\}_{i=1}^{n}$ is a probability distribution over the question words. The values of a_{ti} depend on the hidden state of the decoder at the previous timestep ($\mathbf{s}_{t-1}$), and the hidden state of the question encoder:

$$f(\mathbf{s}_{t-1}, \mathbf{o}_i^{Q^b}) = \mathbf{v}_E^\intercal \tanh(\mathbf{W}_E[\mathbf{s}_{t-1}; \mathbf{o}_i^{Q^b}])$$

where v_E and W_E are learnable parameters.

The attention weights and vectors for the fact encoder are calculated similarly. We use γ_{ti} to refer to the parameterized and normalized attention weights for the fact encoder and $\mathbf{Y}_t^F = \{\gamma_{tj}\}_{j=1}^{j=m}$ is a probability distribution over the fact words. The context vector for the fact can be written as:

$$\mathbf{m}_t = \sum_{i=1}^{m} \gamma_{ti} \mathbf{o}_i^F$$

Decoder: The decoder takes the hidden states from the question and fact encoders to generate the paraphrased sequence of words. Our decoder is a uni-directional LSTM network whose state and context vectors are represented by $\mathbf{s}_t$ and $\mathbf{i}_t$, respectively, such that:

$$\begin{aligned}
\mathbf{i}_t &= [\mathcal{E}^d(p_{t-1}); \mathbf{m}_{t-1}; \mathbf{g}_{t-1}] \\
\mathbf{s}_t &= LSTM(\mathbf{i}_t, \mathbf{s}_{t-1}) \\
\mathbf{s}_0 &= \overleftarrow{h}_1^{Q^b}
\end{aligned}$$

Here $\mathcal{E}^d$ refers to the embedding from the decoder for the paraphrase word, p_{t-1}. The current context and decoder state vectors are combined with the attention vector from the question encoder to obtain the readout state and subsequently the generative probability distribution over the vocabulary using a maxout layer (Goodfellow et al., 2013):

$$\begin{aligned}
\mathbf{r}_t &= \mathbf{W}_r \mathbf{s}_t + \mathbf{U}_r \mathbf{i}_t + \mathbf{V}_r \mathbf{g}_{t-1} \\
\mathbf{Y}_t^V &= \text{softmax}(\mathbf{W}_y \text{ maxout}(\mathbf{r}_t))
\end{aligned}$$

The matrices $\mathbf{W}_y$, $\mathbf{W}_r$, $\mathbf{U}_r$ and $\mathbf{V}_r$ are all learned during training.

Copy mechanism: Recent works for QG handle rare words by employing a pointer network that enables both copying of the words from the input source (answer passage) and generation of words during the decoding process (Gulcehre et al., 2016; See et al., 2017). For question rewriting using facts, we extend this copy mechanism to enable copying from both the input fact words as well as the question words. The copy switch in our case is a softmax function given by

$$\mathbf{p} = \text{softmax}\left(\mathbf{W}_{copy}\mathbf{s}_t + \mathbf{U}_{copy}\mathbf{g}_t + \mathbf{Z}_{copy}\mathbf{m}_t + \mathbf{b}\right)$$

where the matrices $\mathbf{W}_{copy} \in \mathbb{R}^{3\times|s_t|}$, $\mathbf{U}_{copy} \in \mathbb{R}^{3\times|g_t|}$ and $\mathbf{Z}_{copy} \in \mathbb{R}^{3\times|m_t|}$ are learnable parameters and $\mathbf{b}$ is the bias parameter.

During the decoding step, $\mathbf{p}$ is sampled to (1) copy the words from the question, based on $\mathbf{Y}_t^Q$, the normalized, attention weights from the question encoder, or, (2) copy words from the fact based on $\mathbf{Y}_t^F$, the normalized, attention weights from the fact encoder, or (3) generate a new word, based on $\mathbf{Y}_t^V$, the generative distribution on the vocabulary estimated during learning.

4.1 Baselines

Considering the novelty of our proposed task, we are limited in our choice of baselines for comparing with *FIQG*. However, we note that similar to our objective which involves the infusion of parts of an entity fact into a given base question along with possible re-writing of the "wh"-word (for example, "Where" to "Which <location>"), existing QG approaches involve the inclusion of parts of an answer passage into a generated question template using attention and copy mechanisms. Thus, a straightforward application of QG models for our task would involve retraining the model using input passages comprising of both the base question and the fact sentences.

We also highlight that existing paraphrase generation models operate on a source question and generate synonymous variations by substituting specific words with related words and syntactic variations by using other exemplar questions (Fu et al., 2019; Hosking and Lapata, 2021). Consequently, we find QG models more appropriate for fact-infusion and compare *FIQG* with the following state-of-the-art QG baselines:

1. **NQG**[10] is one of the earliest neural seq2seq models proposed for QG using feature-rich input embeddings comprising of words, answer position, parts-of-speech, NER and case information (Zhou et al., 2018).

2. **SGDQG**[11] is a recent model designed to generate complex questions that require reasoning on multiple pieces of information (for example, in the HotpotQA dataset). SGDQG uses semantic graph information constructed from NLP relations between words in the passages (Pan et al., 2020).

3. **GSAQG**[12] uses maxout pointer mechanism

[10] https://github.com/magic282/NQG

[11] https://github.com/YuxiXie/SG-Deep-Question-Generation

[12] https://github.com/seanie12/neural-question-generation

with gated self-attention network to handle long text inputs (Zhao et al., 2018).

4. **RefNet**[13] is a two decoder based model where a second decoder refines the output from the first decoder for generating more complete questions (Nema et al., 2019).

5. **ASs2s**[14] employs an answer-separated seq2seq approach along with a keyword-net and interrogative word identification to handle irrelevant words in generated questions (Kim et al., 2019).

We note that QG models are being widely investigated in current research and some recent innovative aspects in learning QG include the use of variational encoders, graph convolutional networks, and incorporation of global and semantic knowledge (Pan et al., 2020; Wang et al., 2020a; Su et al., 2020; Majumder et al., 2021). Keeping the novelty of our task and dataset in mind, we compare against state-of-the-art models that use components very similar to *FIQG* and defer the investigation of more recent QG research on *FIRS* for future.

5 Experiments and Results

Implementation: We implemented *FIQG* in Python.[15] The hidden state sizes for the two encoders and the decoder are set to 256, whereas the depth for the attention mechanism is set to 512. The readout size is 128 whereas vocabulary size is $\sim 20K$ words, and the target sequence length was set to 50. Dropout rates are set to 0.5 for the dense layers and 0.3 for the attention layers, respectively. A learning rate of 0.001 was used.

Feature-rich embeddings (Zhou et al., 2018) were used for input representations using word, parts-of-speech tags and indicator embeddings. Indicator features using the BIO representation are incorporated in QG models to indicate the answer span in a passage to focus the question around the answer. For our case, this aspect corresponds to the named-entity whose fact we are integrating into the question. However, to differentiate the two cases, namely, when the entity is part of the **a**nswer versus when the entity **n**ame is part of the question, we use an extended set of tags: {BA, IA, BN, IN, O}.

Fact Extraction: The entity descriptions obtained from Google are brief summaries comprising 1-3 long sentences. The crowdworkers however only use specific segments of these summaries or entity facts in their paraphrases. To model this aspect, we used MinIE[16] an unsupervised, domain-independent fact extraction tool on our entity descriptions and mapped a specific fact from the summary with each rewrite (Gashteovski et al., 2017). We provide an example in the Appendix (Footnote 5).

Evaluation: Following existing QG works, we use BLEU (Papineni et al., 2002), METEOR (Lavie and Denkowski, 2009), and ROUGE-L (Lin, 2004) scores to characterize model performance. All three measures are based on calculating the n-gram overlap between human-generated "gold" references and machine-generated predictions.

All baseline models were set up using the configuration settings shared by the authors. As in existing QG studies, we uniformly use pretrained embeddings from GloVe[17] (Pennington et al., 2014) for word representations and tune all models using the BLEU-4 score on the dev portion of the dataset. All experiments were performed on a single GPU on an Nvidia cluster and *FIQG* took approximately 2 hours to train.

5.1 Results and Observations

Fact-Infusion Performance: In Table 5, we summarize the performance of *FIQG* and the baseline models using the different evaluation measures. *FIQG* is able to significantly outperform all baselines on the test data. Even though the number of training instances available in *FIRS* is significantly smaller than datasets such as SQuAD, fact-infused rewriting can be expected to be easier than standard QG since it involves combining a fact with a base question along with potentially rewriting the *wh*-word in contrast with QG where models learn to generate questions for a given passage and an answer-span. Indeed on SQuAD, the state-of-the-art QG models obtain BLEU-4 and METEOR scores about half of that obtained on *FIRS* by *FIQG*. As such, the BLEU scores of the modified QG baselines on *FIRS* are also reasonably high although we note that separately representing the question and fact sentences via the double encoder in *FIQG* results in superior performance over

[13] https://github.com/PrekshaNema25/RefNet-QG

[14] https://github.com/yanghoonkim/NQG_ASs2s

[15] Python 3.7.7, NLTK 3.5, Stanza 1.0.1 libraries were used in feature extraction whereas the deep learning models were implemented in Tensorflow 2.3.0.

[16] https://github.com/uma-pi1/minie

[17] http://nlp.stanford.edu/data/glove.840B.300d.zip

Method	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L
NQG	0.431	0.318	0.247	0.195	0.222	0.484
SGDQG	0.524	0.374	0.278	0.209	0.233	0.482
RefNet	0.567	0.469	0.397	0.338	0.381	0.562
GSAQG	0.572	0.472	0.390	0.322	0.293	0.589
ASs2s	0.614	0.497	0.411	0.342	0.292	0.579
FIQG(Our Model)	**0.729**	**0.623**	**0.547**	**0.486**	**0.382**	**0.686**
			Ablation Experiments			
-GloVe	0.634	0.510	0.429	0.367	0.331	0.623
-Indicator Features	0.721	0.608	0.528	0.464	0.376	0.675
-POS Features	0.705	0.598	0.523	0.463	0.371	0.677
w/ Combined Indicator	0.717	0.608	0.531	0.469	0.376	0.676

Table 5: Question Paraphrase Generation Results on *FIRS*

the baselines. Indeed, statistically significant gains are seen on all evaluation measures except the METEOR score for which the performance is similar to that of **RefNet**.

Ablation Experiments: The results of our ablation experiments are also shown in Table 5. Not surprisingly, and as shown in other QG studies, initializing our word embeddings with pretrained embeddings results in improved question rewriting performance. Without initialization from GloVe vectors, we observe a significant drop in the scores. Similarly, indicator features are known to help QG by providing signals to the model on what parts of the passage the generation should be focused on. For our smaller sentences, excluding them yields a small drop in performance. Moreover, discriminating between the two cases (answer versus question entity) seems to help the model attain a minor improvement in performance over using a single set of indicators as shown in the 'w/ Combined Indicator" row of Table 5. Although as observed in Section 3.1, the extra "fact" words in rewrites are often nouns and words related to nouns, excluding POS tag information causes a small drop in the performance. Based on these results, we can attribute the overall performance of *FIQG* mostly to the network architecture coupled with appropriately initialized word embedding features.

Anecdotal observations: We show sample test predictions with *FIQG* in Table 6 for discussion. In the first example, a fact related to an entity mentioned in the question ("Martin Luther") is being utilized whereas in the second example, the fact relates to "David Booth", the answer entity. *FIQG* missed some words from the human-specified variation ("target") in the first case and gets the tense wrong in the second example. However, we note that the fact extracted from the summary did not contain the extra initials whereas the tense is also specified incorrectly in the base question from SQuAD. Barring these minor aspects, the predictions are legitimate and complete and in the second example we also note the change in the wh-word.

Base Question: When did `Martin Luther` publish his translation of the New Testament? **Entity Description**: Martin Luther, O.S.A. was a German professor of theology, composer, priest, Augustinian monk, and a seminal figure in the Protestant Reformation. Martin Luther was ordained to the priesthood in 1507. **Fact**: Martin Luther was ordained to the priesthood in 1507. **Target**: When did Martin Luther, O.S.A., who was ordained to the priesthood in 1507, publish his translation of the New Testament? **Prediction**: when did martin luther, *ordained to priesthood 1507*, publish his translation of the new testament ?
Base question: Who decide to make a very large donation to the university's Booth School of Business? **Entity Description**: `David Gilbert Booth` is an American businessman, investor, and philanthropist. He is the Executive Chairman of Dimensional Fund Advisors, which he co-founded with Rex Sinquefield. **Fact**: David Gilbert Booth is American businessman **Target**: What American businessman decided to make a very large donation to the university's Booth School of Business? **Prediction**: what *american businessman* decide to make a very large donation to the university 's booth school of business ?

Table 6: Anecdotes from *FIQG* predictions

6 Conclusions and Future Work

We presented *FIRS*, to the best of our knowledge, a first-of-its-kind dataset containing fact-infused variations of a subset of questions from SQuAD. We proposed a double encoder-decoder model *FIQG*, for learning to generate question variations through fact infusion. *FIQG* is able to significantly outperform extensions of standard QG models on *FIRS*.

In future, we would like to investigate the use of question variations on downstream tasks such as QA, reading comprehension, and interactive dialog (Tang et al., 2018; Ribeiro et al., 2019; Gao et al., 2020). Additionally, question variations available in *FIRS* can be used for learning diverse question generation, adversarial models for QA, and QG on multiple passages (Ren et al., 2018; Yang et al., 2018; Gan and Ng, 2019). We would like to explore these aspects as well as study novel learning methods such as variational auto-encoders and reinforcement learning for improving performance on the fact-infused question generation task (Misra et al., 2018; Li et al., 2018).

Ethics Statement

This research was conducted in conformance with the ACM Code of Ethics.

Acknowledgements

This research is supported by the National Research Foundation, Singapore under its Industry Alignment Fund – Pre-positioning (IAF-PP) Funding Initiative. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore.

References

Vamshi Ambati. 2008. Dependency structure trees in syntax based machine translation. In *Adv. MT Seminar Course Report. Vol 137*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*.

Soumen Chakrabarti. 2020. Interpretable complex question answering. In *Proceedings of The Web Conference 2020*, page 2455–2457.

Nan Duan, Duyu Tang, Peng Chen, and Ming Zhou. 2017. Question generation for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 866–874.

Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2013. Paraphrase-driven learning for open question answering. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1608–1618.

Yao Fu, Yansong Feng, and John P Cunningham. 2019. Paraphrase generation with latent bag of words. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Wee Chung Gan and Hwee Tou Ng. 2019. Improving the robustness of question answering systems to question paraphrasing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6065–6075.

Silin Gao, Yichi Zhang, Zhijian Ou, and Zhou Yu. 2020. Paraphrase augmented task-oriented dialog generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 639–649.

Kiril Gashteovski, Rainer Gemulla, and Luciano del Corro. 2017. MinIE: Minimizing facts in open information extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2630–2640.

Ian J. Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. 2013. Maxout networks. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, page III–1319–III–1327. JMLR.org.

Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the unknown words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Ankush Gupta, Arvind Agarwal, Prawaan Singh, and Piyush Rai. 2018. A deep generative framework for paraphrase generation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 5149–5156.

Vrindavan Harrison and Marilyn Walker. 2018. Neural generation of diverse questions using answer focus, contextual and linguistic features. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 296–306.

Michael Heilman and Noah A. Smith. 2010. Good question! statistical ranking for question generation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 609–617.

Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, and Gerhard Weikum. 2013. Yago2: A spatially and temporally enhanced knowledge base from wikipedia. *Artif. Intell.*, 194:28–61.

Tom Hosking and Mirella Lapata. 2021. Factorising meaning and form for intent-preserving paraphrasing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*.

Minlie Huang, Xiaoyan Zhu, and Jianfeng Gao. 2020. Challenges in building intelligent open-domain dialog systems. *ACM Trans. Inf. Syst.*, 38(3).

Xiao Huang, Jingyuan Zhang, Dingcheng Li, and Ping Li. 2019. Knowledge graph embedding based question answering. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, WSDM '19, page 105–113. Association for Computing Machinery.

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

Kettip Kriangchaivech and Artit Wangperawong. 2019. Question generation by transformers. *CoRR*, abs/1909.05017.

Florian Kunneman, Thiago Castro Ferreira, Emiel Krahmer, and Antal van den Bosch. 2019. Question similarity in community question answering: A systematic exploration of preprocessing methods and models. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 593–601.

Alon Lavie and Michael J. Denkowski. 2009. The meteor metric for automatic evaluation of machine translation. In *Machine Translation*.

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.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. 2015. DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal*, 6(2):167–195.

Zichao Li, Xin Jiang, Lifeng Shang, and Hang Li. 2018. Paraphrase generation with deep reinforcement learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3865–3878.

Chen Liang, Praveen Paritosh, Vinodh Rajendran, and Kenneth D. Forbus. 2016. Learning paraphrase identification with structural alignment. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, page 2859–2865.

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

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

Bodhisattwa Prasad Majumder, Sudha Rao, Michel Galley, and Julian McAuley. 2021. Ask what's missing and what's useful: Improving clarification question generation using global knowledge. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4300–4312.

Ishan Misra, Ross Girshick, Rob Fergus, Martial Hebert, Abhinav Gupta, and Laurens van der Maaten. 2018. Learning by Asking Questions. In *CVPR*.

Preksha Nema and Mitesh M. Khapra. 2018. Towards a better metric for evaluating question generation systems. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3950–3959.

Preksha Nema, Akash Kumar Mohankumar, Mitesh M. Khapra, Balaji Vasan Srinivasan, and Balaraman Ravindran. 2019. Let's ask again: Refine network for automatic 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 3314–3323.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches*, volume 1773.

Şaziye Betül Özateş, Arzucan Özgür, and Dragomir Radev. 2016. Sentence similarity based on dependency tree kernels for multi-document summarization. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2833–2838, Portorož, Slovenia.

Liangming Pan, Wenqiang Lei, Tat-Seng Chua, and Min-Yen Kan. 2019. Recent advances in neural question generation. *CoRR*, abs/1905.08949.

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 *Proceedings of the*

40th Annual Meeting of the Association for Computational Linguistics, pages 311–318.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

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.

Yuanhang Ren, Ye Du, and Di Wang. 2018. Tackling adversarial examples in QA via answer sentence selection. In *Proceedings of the Workshop on Machine Reading for Question Answering*, pages 31–36.

Marco Tulio Ribeiro, Carlos Guestrin, and Sameer Singh. 2019. Are red roses red? evaluating consistency of question-answering models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6174–6184, Florence, Italy. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083.

Heung-yeung Shum, Xiao-dong He, and Di Li. 2018. From eliza to xiaoice: challenges and opportunities with social chatbots. In *Frontiers of Information Technology & Electronic Engineering*.

Tim Steuer, Anna Filighera, and Christoph Rensing. 2020. Remember the facts? investigating answer-aware neural question generation for text comprehension. In *Artificial Intelligence in Education*.

Dan Su, Yan Xu, Wenliang Dai, Ziwei Ji, Tiezheng Yu, and Pascale Fung. 2020. Multi-hop question generation with graph convolutional network. In *Findings of the Association for Computational Linguistics: EMNLP 2020*.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, page 3104–3112.

A. Talmor and J. Berant. 2018. The web as a knowledge-base for answering complex questions. In *North American Association for Computational Linguistics (NAACL)*.

Duyu Tang, Nan Duan, Zhao Yan, Zhirui Zhang, Yibo Sun, Shujie Liu, Yuanhua Lv, and Ming Zhou. 2018. Learning to collaborate for question answering and asking. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1564–1574.

Luu Anh Tuan, Darsh J. Shah, and Regina Barzilay. 2020. Capturing greater context for question generation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence*, pages 9065–9072.

Siyuan Wang, Zhongyu Wei, Zhihao Fan, Zengfeng Huang, Weijian Sun, Qi Zhang, and Xuanjing Huang. 2020a. PathQG: Neural question generation from facts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Su Wang, Rahul Gupta, Nancy Chang, and Jason Baldridge. 2019. A task in a suit and a tie: Paraphrase generation with semantic augmentation. In *The Thirty-Third AAAI Conference on Artificial Intelligence*, pages 7176–7183.

Xun Wang, Katsuhito Sudoh, Masaaki Nagata, Tomohide Shibata, Daisuke Kawahara, and Sadao Kurohashi. 2016. Reading comprehension using entity-based memory network. *arXiv preprint arXiv:1612.03551*.

Zhen Wang, Siwei Rao, Jie Zhang, Zhen Qin, Guangjian Tian, and Jun Wang. 2020b. Diversify question generation with continuous content selectors and question type modeling. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2134–2143, Online. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380.

Shiqi Zhao and Haifeng Wang. 2010. Paraphrases and applications. In *Coling 2010: Paraphrases and Applications–Tutorial notes*, pages 1–87, Beijing, China.

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

Qingyu Zhou, Nan Yang, Furu Wei, Chuanqi Tan, Hangbo Bao, and Ming Zhou. 2018. Neural question generation from text: A preliminary study. In *Natural Language Processing and Chinese Computing*, pages 662–671, Cham. Springer International Publishing.