MixQG: Neural Question Generation with Mixed Answer Types

Lidiya Murakhovs'ka **Chien-Sheng Wu** **Philippe Laban**
Tong Niu **Wenhao Liu** **Caiming Xiong**
Salesforce AI Research
{l.murakhovska, wu.jason, plaban, tniu, wenhao.liu, cxiong}@salesforce.com

Abstract

Asking good questions is an essential ability for both human and machine intelligence. However, existing neural question generation approaches mainly focus on short factoid type of answers. In this paper, we introduce a neural question generator, MixQG, to bridge this gap. We combine nine question answering datasets with diverse answer types, including yes/no, multiple-choice, extractive, and abstractive answers, to train a single generative model. We show with empirical results that our model outperforms existing work in both seen and unseen domains, and can generate questions with different cognitive levels when conditioned on different answer types. We run a human evaluation study to assess the quality of generated questions and find that MixQG outperforms the next best model by 10%. Our code and model checkpoints will be released and integrated with the HuggingFace library to facilitate various downstream applications.

1 Introduction

Question generation (QG) aims to automatically create questions from a given text passage or document with or without answers. It has a wide range of applications such as improving question answering (QA) systems (Duan et al., 2017) and search engines (Han et al., 2019) through data augmentation, making chatbots more engaging (Wang et al., 2018; Laban et al., 2020), enabling automatic evaluation (Rebuffel et al., 2021) and fact verification (Pan et al., 2021), and facilitating educational applications (Chen et al., 2018).

Earlier QG approaches relied on syntactic rules that incorporated linguistic features into the QG process (Heilman and Smith, 2010; Khullar et al., 2018). Du et al. (2017) pointed out some of the limitations of such rule-based systems and formulated the task of question generation as a sequence-to-sequence learning problem. Based on this formulation, recent works rely on pre-trained Transformer-based models to generate answer-aware questions (Dong et al., 2019a; Yan et al., 2020a; Lelkes et al., 2021). However, the majority of QG research so far has been performed on the SQuAD dataset (Rajpurkar et al., 2016), and as a result, it mainly focuses on factoid short answer questions (Zhang and Bansal, 2019; Zhou et al., 2019; Su et al., 2020).

Context: In the late 17th century, Robert Boyle proved that air is necessary for combustion. English chemist John Mayow (1641–1679) refined this work by showing that fire requires only a part of air that he called spiritus nitroaereus or just nitroaereus. In one experiment he found that placing either a mouse or a lit candle in a closed container over water caused the water to rise and replace one-fourteenth of the air's volume before extinguishing the subjects. From this he surmised that nitroaereus is consumed in both respiration and combustion.
Question: Who proved that air is necessary for combustion? **Ext. Short Answer**: Robert Boyle
Question: How did John Mayow find that spiritus nitroaereus is consumed in both respiration and combustion? **Abs. Short Answer**: through an experiment
Question: Does fire need air to burn? **Yes/No Answer**: yes
Question: What did John Mayow discover about nitroaereus? **Ext. Long Answer**: In the late 17th century ... in both respiration and combustion.
Question: Why was the mouse used in the experiment? **Abs. Long Answer**: The mouse was used in the experiment to test the consumption of nitroaereus during respiration.

Figure 1: Given the same context, MixQG generates diverse questions based on the target answer choice.

In reality, answers can come in a variety of types and forms, e.g., short/long, multiple-choice, yes-no, and extractive/abstractive answers. We hypothesize that *answer types are as important as question types*, and that different answer types have their unique QG challenges and result in questions with different cognitive levels. MixQG combines nine QA datasets with varied answer types to build a more robust and versatile QG model. We use pre-trained generative language models like T5 (Raffel et al., 2020) and BART (Lewis et al., 2019) without question-specific or domain-specific prefixes to generate the questions. Figure 1 illustrates the

Dataset	Type	Source	Train examples	Dev. examples
SQuAD (Rajpurkar et al., 2016)	Extractive	Wikipedia	86,588	10,507
NewsQA (Trischler et al., 2017)	Extractive	News	74,160	4,212
TriviaQA (Joshi et al., 2017)	Extractive	Web	61,688	7,785
SearchQA (Dunn et al., 2017)	Extractive	Web	117,384	16,980
HotpotQA (Yang et al., 2018)	Extractive	Wikipedia	72,928	5,904
NQ (Kwiatkowski et al., 2019)	Extractive	Wikipedia	104,071	12,836
NarQA (Kočiský et al., 2018)	Abstractive	Wikipedia, Project Gutenberg	32,747	3,461
MCTest (Richardson et al., 2013)	Multiple-Choice	Stories	1,200	600
BoolQ (Clark et al., 2019)	Yes-No	Wikipedia	9,427	3,270
Quoref* (Dasigi et al., 2019)	Extractive	Wikipedia	19,399	2,418
QAConv* (Wu et al., 2021)	Extractive	Email, Panel, Channel	25,988	3,251
DROP* (Dua et al., 2019)	Abstractive	Wikipedia	77,400	9,535
TweetQA* (Xiong et al., 2019)	Abstractive	Twitter	10,692	1,086

Table 1: Dataset Statistics of various QA corpora. * indicates unseen corpus during training.

above, showing MixQG-generated questions of different cognitive levels for different answer types.

The contribution of this paper is summarized as follows: 1) We train a unified QG model that achieves state-of-the-art performance in both seen and unseen domains. We release training code and model checkpoints (base, large, 3B) to facilitate various downstream QG applications [1]. 2) We show that MixQG is able to produce different cognitive level questions by controlling the answer types. We conduct a human evaluation study which confirms that MixQG leads to improvements in question quality in a practical quiz design setting.

2 Methodology

2.1 Datasets

We leverage nine commonly used QA datasets (Table 1) to train our MixQG model, including six MRQA 2019 Shared Task (Fisch et al., 2019) datasets, NarrativeQA (Kočiský et al., 2018), MCTest (Richardson et al., 2013), and BoolQ (Clark et al., 2019). These represent the majority of large-scale publicly available QA datasets. We obtain in total 560,193 training examples with different answer types and source domains. We reserve their validation set for in-domain evaluation.

In most general sense, a QA dataset comprises of <C, Q, A> tuples, where C is a context document, Q is a human-written question, and A is its corresponding answer. Following a common classification of answer types, we bucket each dataset into one of the below categories: 1) **Extractive [EX]**: the answer to the question is a substring of the context passage. 2) **Abstractive [AB]**: the answer to the question is written in free-form and is not necessarily contained within the context passage. 3) **Multiple-Choice [MC]**: question comes with multiple answers to select from, including a single correct option and several distractors. 4) **Yes-No [YN]**: the answer is a boolean response. Datasets that do not comply with the above format, such as ELI5 (Fan et al., 2019) and GooAQ (Khashabi et al., 2021), were excluded from training. We leave their exploration to future work.

We also leverage a set of datasets unseen during training to evaluate our model's generalization ability. Similar to the train datasets, these cover several text sources, domains, and answer types. Quoref (Dasigi et al., 2019) questions can have disjoint spans as answers and often require coreference resolution. DROP (Dua et al., 2019) questions require discrete reasoning over the context paragraphs. QAConv (Wu et al., 2021) uses informative conversations such as emails, channels, and panels as a knowledge source, and it includes extractive answers from multiple text spans. TweetQA (Xiong et al., 2019) uses social media as an information source and contains abstractive answers.

Note that to generate fluent questions, we need to place some restrictions on the training data we use. For example, we disregard "fill-in-the-blank" (a.k.a Cloze-style) reading comprehension datasets as their questions are implicit and thus do not aid the QG model. Similarly, we ensure that our training data does not contain unanswerable questions or multiple-choice questions that are too general (e.g., “which of the following is TRUE according to the passage?”).

[1] https://github.com/salesforce/QGen

Type	Input
EX	{answer} \n {context}
AB	{answer} \n {context}
MC	{correct_answer} \n {context}
YN	{answer} + {entities} \n {context}

Table 2: Input answer formatting.

2.2 Language Modeling

We rely on a text-to-text framework as a basis for MixQG (Training details are in Section A). When combining our training datasets, we encode all inputs and outputs into a unified plain-text format. For answer-aware question generation, the input is usually formatted in one of the two ways: (1) prepending (**-pre**) the answer before the context and separating it from the rest of the text by a special separator token or (2) highlighting (**-hl**) the answer span within the context with special highlight tokens (Chan and Fan, 2019). To maintain flexibility, we rely on prepending the answer since highlighting is only applicable to the extractive answer types. In particular, we format the inputs to our model such that the answer always precedes the context paragraph and use a "\n" separator in between, as shown in Table 2.

For MC type of data, we only take the correct answer and disregard the distractor options. For YN data, we extract entities from the question using spaCy's NER model [2] and append them to the answer. The reason for adding additional entities is to restrict the domain of questions, as given a context paragraph, there are many boolean questions whose answer would be yes or no, without further restriction. Note that no type-specific prefixes are added to the input representation, and the corresponding questions are used as output.

3 Experimental Results

3.1 Automatic Metrics

We report the commonly-used metrics applied in the QG research: BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), and METEOR (Banerjee and Lavie, 2005) scores. We also report BERTScore (Zhang et al., 2020), which relies on contextual embeddings to produce the final score.

3.2 In-Domain Analysis

In Table 3, we compare baselines trained solely on the target in-domain dataset against MixQG and MixQG$_{finetuned}$. MixQG indicates our model that is joint trained on nine QA datasets with random sampling, and MixQG$_{finetuned}$ is the one further fine-tuned on the target dataset. We show results on two datasets: SQuAD and NQ. Since SQuAD is the most common benchmark for QG, we additionally compare MixQG against existing question generation models such as ProphetNet (Qi et al., 2020) and other T5 variants. The results show that MixQG outperforms an equally sized model trained directly on the target dataset. Given that question styles and dataset domains may vary across MixQG's seed datasets, additional fine-tuning on the target dataset further improves the scores. This shows that MixQG is a strong pretrained model which can be further adapted to specific use cases.

3.3 Out-of-Domain Analysis

Table 4 summarizes the evaluations on out-of-domain datasets of extractive and abstractive answer types. We observe that a dedicated model trained on the target dataset outperforms MixQG in a zero-shot setting. One potential reason is that answer and question style in different QA datasets may differ significantly. For example, answers are ambiguous pronouns in the Quoref dataset, and questions in DROP dataset are intentionally created for discrete reasoning. However, MixQG$_{finetuned}$ obtains the best overall scores after further fine-tuning on the target training set, suggesting that MixQG is a strong starting point for further fine-tuning question generation models.

3.4 Human Evaluation

Recent studies have shown that n-gram based metrics may not correlate well with human judgements Nema and Khapra (2018). The objective of human evaluation is to evaluate QG models by measuring how useful they are as a tool to aid teachers in quiz creation. We compare seven QG models and collect 3,164 human-annotated samples from 10 recruited teachers. More details are in Section B.

Quiz Design Task Given an article on the quiz topic selected from Wikipedia, teachers are asked to specify a quiz concept (a subset of the article) they want to test their students on. This is used as the target answer input for QG models. Teachers can then approve a generated question to be included on the quiz or reject it and provide a reason for rejection. The success of a QG model depends on its question approval rate.

Besides MixQG, three GPT2 baselines (Radford

[2]https://spacy.io/api/entityrecognizer

Dataset	Model	Size	BLEU	R1	R2	RL	RLsum	METEOR	BERTScore
SQuAD	ProphetNet-pre	large	22.88	51.37	29.48	47.11	47.09	41.46	0.4931
	BART-hl	base	21.13	51.88	29.43	48.00	48.01	40.23	0.5433
	T5-hl	base	23.19	53.52	31.22	49.40	49.40	42.68	0.5548
	BART-pre	base	22.09	52.75	30.56	48.79	48.78	41.39	0.5486
	T5-pre	base	**23.74**	54.12	31.84	49.82	49.81	43.63	0.5568
	MixQG	base	23.53	54.39	32.06	50.05	50.02	43.83	0.5566
	MixQG$_{finetuned}$	base	23.46	**54.48**	**32.18**	**50.14**	**50.10**	**44.15**	**0.5582**
	MixQG	3B	**25.42**	**56.11**	**33.91**	**51.85**	**51.86**	**45.75**	**0.5789**
NQ	T5-pre	base	29.99	59.53	37.83	56.65	56.64	54.38	0.5202
	MixQG	base	30.69	60.04	38.43	57.09	57.09	54.76	0.5246
	MixQG$_{finetuned}$	base	**31.25**	**60.98**	**39.21**	**57.84**	**57.84**	**55.90**	**0.5351**
	MixQG	3B	**33.91**	**63.17**	**41.95**	**60.15**	**60.15**	**58.34**	**0.5610**

Table 3: Results on two seen datasets, SQuAD (Rajpurkar et al., 2016) and NQ (Kwiatkowski et al., 2019).

Answer Type	Dataset	Model	BLEU	R1	R2	RL	RLsum	METEOR	BERTScore
EX	QAConv	T5-pre	21.32	45.94	27.91	42.92	42.90	38.27	0.4374
		MixQG	16.65	39.99	22.01	37.62	37.59	29.07	0.4117
		MixQG$_{finetuned}$	**22.74**	**47.40**	**29.48**	**44.41**	**44.40**	**39.93**	**0.4533**
EX	Quoref	T5-pre	26.88	45.54	31.98	44.10	44.12	41.84	**0.4150**
		MixQG	4.28	24.89	7.97	22.27	22.30	14.13	0.2859
		MixQG$_{finetuned}$	**27.36**	**45.91**	**32.41**	**44.42**	**44.42**	**42.06**	0.4137
AB	DROP	T5-pre	28.46	53.48	35.49	50.97	51.00	47.50	0.5491
		MixQG	7.16	30.66	12.95	28.38	28.40	23.23	0.3556
		MixQG$_{finetuned}$	**28.53**	**53.72**	**35.63**	**51.11**	**51.12**	**47.83**	**0.5493**
AB	TweetQA	T5-pre	17.02	45.28	23.28	44.20	44.18	44.63	0.4384
		MixQG	5.28	28.18	10.65	26.91	26.89	28.83	0.2653
		MixQG$_{finetuned}$	**18.66**	**47.12**	**24.95**	**45.97**	**45.94**	**46.60**	**0.4645**

Table 4: Results on unseen datasets, QAConv (Wu et al., 2021), Quoref (Dasigi et al., 2019), DROP (Dua et al., 2019), and TweetQA (Xiong et al., 2019). All models are of size base.

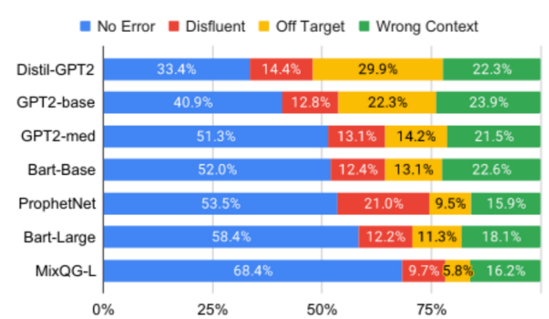

Figure 2: Human approval rate of seven QG models.

et al., 2019), two BART baselines (Lewis et al., 2019), and ProphetNet-Large finetuned on SQuAD are evaluated. In Figure 2, we see that MixQG attains a 68.4% acceptance rate, outperforming the next best model by 10%. MixQG also generates the smallest number of disfluent and off target (answer mismatch) questions - with majority of errors coming from wrong context (too general or too specific) questions. Generating questions with the right level of specificity remains a challenge and is a promising direction for future work.

3.5 Qualitative Analysis

First, we compare MixQG generated questions to the gold questions annotated in five public QA datasets (Table 5). We find that the generated questions are fluent, relevant, and reasonable to the provided answer and context, even if they differ from the gold label. This further motivates the need of human evaluation for QG research.

Second, we use the HuggingFace summarization pipeline to obtain the summary of the context, and we feed each sentence of the summary as the target answer to MixQG to obtain questions. In this way, we can test MixQG's generalization ability to abstractive answers. As shown in Figure 5, we observe that feeding in long and abstractive answers can still generate fluent and reasonable questions, suggesting that it is possible to control the question's cognitive level by its answer. We leave as future work further research into summary-based unsupervised QA-pair generation.

Lastly, in the Quiz Design study, we find there are 106 cases in which the teachers only accepted a single candidate question into the quiz. MixQG produced the accepted candidate 47 times, more than any of the other models. We provide three examples of such MixQG-only success cases as well as three instances in which the MixQG's question was not accepted in Table 6.

4 Related Work

Question generation's practical importance has lead to an increasing interest in the field. The early work in QG relied on linguistic templates and rules to produce questions from declarative sentences (Heilman and Smith, 2010; Labutov et al., 2015). With the success of neural techniques in text generation tasks, applying neural sequence-to-sequence generation models became more common (Du et al., 2017; Sun et al., 2018). More recent works leverage pre-trained transformer based networks, such as T5 (Raffel et al., 2020), BART (Lewis et al., 2019), PEGASUS (Zhang et al., 2019) and ProphetNet (Yan et al., 2020b), for question generation which have been successful in many applications (Dong et al., 2019b; Lelkes et al., 2021; Rebuffel et al., 2021; Pan et al., 2021).

However, most of the earlier work focuses on using a single QA dataset, such as SQuAD (Rajpurkar et al., 2016). While working on generation of open-ended (Cao and Wang, 2021), controllable (Cao and Wang, 2021), multi-hop (Cho et al., 2021) or cause-effect (Stasaski et al., 2021) questions has gained attention, each direction is studied in isolation as it usually requires a separate QA dataset.

Most directly related to our work is UnifiedQA (Khashabi et al., 2020), which successfully crosses format boundaries of different QA datasets to train a robust QA system. It advocates for more general and broader system designs not limited to specific dataset formats. Similar to their approach, MixQG combines multiple QA datasets and trains a single QG system in a text-to-text paradigm.

5 Conclusion

In this paper, we present MixQG, a question generation model pre-trained on a collection of QA datasets with a mix of answer types. We show through experiments that the resulting model is a strong starting point for further fine-tuning which achieves state-of-the-art results on target datasets in commonly-used similarity metrics as well as our designed human evaluation. We release our code and the model checkpoints to facilitate QG research and downstream applications.

6 Ethical Considerations

MixQG is subject to biases found in the training data of both the underlying text-to-text models and all QA datasets that we have used for pre-training. We do not collect a new dataset for question generation and instead reuse data from previously published works. As such, we rely on the published works to follow the responsible data collection practices. The model is currently English language only which limits its practical applications in the real world. We hope to make MixQG multilingual as more diverse QA datasets become available in the future. We validate the proposed model by conducting a human evaluation. We recruited 10 teachers for a study that lasted a maximum of two hours and gifted each participant a $50 gift card.

References

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

Shuyang Cao and Lu Wang. 2021. Controllable open-ended question generation with A new question type ontology. *CoRR*, abs/2107.00152.

Ying-Hong Chan and Yao-Chung Fan. 2019. A recurrent BERT-based model for question generation. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 154–162, Hong Kong, China. Association for Computational Linguistics.

Guanliang Chen, Jie Yang, Claudia Hauff, and Geert-Jan Houben. 2018. Learningq: a large-scale dataset for educational question generation. In *Twelfth International AAAI Conference on Web and Social Media*.

Woon Sang Cho, Yizhe Zhang, Sudha Rao, Asli Celikyilmaz, Chenyan Xiong, Jianfeng Gao, Mengdi Wang, and Bill Dolan. 2021. Contrastive multi-document question generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 12–30, Online. Association for Computational Linguistics.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American*

Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Pradeep Dasigi, Nelson F. Liu, Ana Marasović, Noah A. Smith, and Matt Gardner. 2019. Quoref: A reading comprehension dataset with questions requiring coreferential reasoning. In *Proc. of EMNLP-IJCNLP*.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019a. Unified language model pre-training for natural language understanding and generation. *CoRR*, abs/1905.03197.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019b. Unified language model pre-training for natural language understanding and generation. In *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. *CoRR*, abs/1705.00106.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proc. of NAACL*.

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, Copenhagen, Denmark. Association for Computational Linguistics.

Matthew Dunn, Levent Sagun, Mike Higgins, V. Ugur Güney, Volkan Cirik, and Kyunghyun Cho. 2017. Searchqa: A new q&a dataset augmented with context from a search engine. *CoRR*, abs/1704.05179.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy. Association for Computational Linguistics.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. 2019. MRQA 2019 shared task: Evaluating generalization in reading comprehension. *CoRR*, abs/1910.09753.

Fred.X Han, Di Niu, Kunfeng Lai, Weidong Guo, Yancheng He, and Yu Xu. 2019. Inferring search queries from web documents via a graph-augmented sequence to attention network. In *The World Wide Web Conference*, WWW '19, page 2792–2798, New York, NY, USA. Association for Computing Machinery.

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, Los Angeles, California. Association for Computational Linguistics.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. UNIFIEDQA: Crossing format boundaries with a single QA system. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1896–1907, Online. Association for Computational Linguistics.

Daniel Khashabi, Amos Ng, Tushar Khot, Ashish Sabharwal, Hannaneh Hajishirzi, and Chris Callison-Burch. 2021. GooAQ: Open question answering with diverse answer types. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 421–433, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Payal Khullar, Konigari Rachna, Mukul Hase, and Manish Shrivastava. 2018. Automatic question generation using relative pronouns and adverbs. In *Proceedings of ACL 2018, Student Research Workshop*, pages 153–158, Melbourne, Australia. Association for Computational Linguistics.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The NarrativeQA reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Philippe Laban, John Canny, and Marti A Hearst. 2020. What's the latest? a question-driven news chatbot. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 380–387.

Igor Labutov, Sumit Basu, and Lucy Vanderwende. 2015. Deep questions without deep understanding.

In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 889–898, Beijing, China. Association for Computational Linguistics.

Adam D. Lelkes, Vinh Q. Tran, and Cong Yu. 2021. Quiz-Style Question Generation for News Stories. In *Proc. of the the Web Conf. 2021*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461.

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

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101.

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.

Liangming Pan, Wenhu Chen, Wenhan Xiong, Min-Yen Kan, and William Yang Wang. 2021. Zero-shot fact verification by claim generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 476–483, Online. Association for Computational Linguistics.

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, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Weizhen Qi, Yu Yan, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. 2020. ProphetNet: Predicting future n-gram for sequence-to-SequencePre-training. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2401–2410, Online. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

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

Clément Rebuffel, Thomas Scialom, Laure Soulier, Benjamin Piwowarski, Sylvain Lamprier, Jacopo Staiano, Geoffrey Scoutheeten, and Patrick Gallinari. 2021. Data-questeval: A referenceless metric for data to text semantic evaluation. *arXiv preprint arXiv:2104.07555*.

Matthew Richardson, Christopher J.C. Burges, and Erin Renshaw. 2013. MCTest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 193–203, Seattle, Washington, USA. Association for Computational Linguistics.

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

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*, pages 4636–4647, Online. Association for Computational Linguistics.

Xingwu Sun, Jing Liu, Yajuan Lyu, Wei He, Yanjun Ma, and Shi Wang. 2018. Answer-focused and position-aware neural question generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3930–3939, Brussels, Belgium. Association for Computational Linguistics.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. NewsQA: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200, Vancouver, Canada. Association for Computational Linguistics.

Yansen Wang, Chenyi Liu, Minlie Huang, and Liqiang Nie. 2018. Learning to ask questions in open-domain conversational systems with typed decoders. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2193–2203, Melbourne, Australia. Association for Computational Linguistics.

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

Chien-Sheng Wu, Andrea Madotto, Wenhao Liu, Pascale Fung, and Caiming Xiong. 2021. Qaconv: Question answering on informative conversations. *arXiv preprint arXiv:2105.06912*.

Wenhan Xiong, Jiawei Wu, Hong Wang, Vivek Kulkarni, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. 2019. TWEETQA: A social media focused question answering dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5020–5031, Florence, Italy. Association for Computational Linguistics.

Yu Yan, Weizhen Qi, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. 2020a. Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training. *arXiv preprint arXiv:2001.04063*.

Yu Yan, Weizhen Qi, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. 2020b. Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training. *CoRR*, abs/2001.04063.

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, Brussels, Belgium. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2019. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. *CoRR*, abs/1912.08777.

Shiyue Zhang and Mohit Bansal. 2019. Addressing semantic drift in question generation for semi-supervised question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2495–2509, Hong Kong, China. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

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

A Training Details

Training datasets are listed in Table 1. For training MixQG, we use several pre-trained text-to-text model checkpoints from the HuggingFace library (Wolf et al., 2020). We finetune them for question generation using our combined dataset described in Section 2.1. For most experiments done in this paper, we finetune on a T5-base model (Raffel et al., 2020). We also scale up the model and report results for T5-large, T5-3B, and BART-large settings (Appendix D). We train for 100,000 steps (or 22 epochs) with a learning rate of 3×10^{-5} using the AdamW (Loshchilov and Hutter, 2017) optimizer and a batch size of 32. All training was done on eight A100 NVIDIA GPUs and took approximately 35 hours.

B Quiz Design Task Details

We recruit teachers or ex-teachers from an online group forum. In total, 20 participants filled out the interest form, 14 were selected, and 10 completed the study. The participants had been teachers for at least a year and 3.6 years on average, and had taught diverse subjects such as sciences, history, literature, and IT topics, at various levels from primary school to college-level. The study was meant to last a maximum of two hours, and participants were gifted a $50 gift card upon completion.

Participants were tasked with creating between 5-7 quizzes, each with a minimum of 8 concepts, and could pick from a set list of 7 quiz topics, which we pre-selected from the list of featured Wikipedia articles[3]. We purposefully selected articles within different domains to benchmark the QGen models in diverse topical settings: two in physics (Sustainable Energy, Californium Atom), two in biology (DNA, Enzymes), two in history (Statue of Liberty, Palazzo Pitti), and one in geology (the K-T extinction). Participants were given the first 500 words

[3]https://en.wikipedia.org/wiki/Wikipedia:Featured_articles

of the Wikipedia page of each topic as reading material to select Quiz concepts from. User interface is shown in Figure 4. Hierarchical categorization of errors for question generation is shown in Figure 3.

C Qualitative Study Details

To understand MixQG's performance beyond automated metrics, we analyze its generated questions in Table 5. It shows several examples of questions generated by MixQG-3B on the validation sets of different datasates along with the ground-truth questions. We also generate question-answer pairs on Wikipedia articles using a pipeline approach as shown in Figure 5. First, we use a summarization model [4] to obtain the summary of the context. Then we feed each sentence of the summary as the target answer to MixQG and obtain the questions. We observe that the generated questions are grammatically fluent, relevant to the input, and answerable by the target answer paragraph. We find that feeding in longer answers to the model generates more general, higher-level questions about the source article, while short answers prompt more factoid-style questions. As a result, we are able to generate questions of varied cognitive levels from the same source document by restricting the answer part of the input.

D Scaling

Table 7 shows the performance of differently sized MixQG models on SQuAD dataset. We additionally train MixQG model based on BART-large checkpoint, referred to as MixQG$^{BART}_{large}$. As expected, the largest MixQG model (3 billion parameters) performs best among the different model size variants.

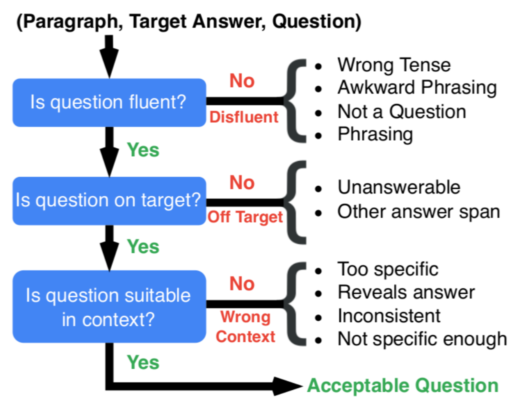

Figure 3: Hierarchical categorization of errors for question generation. Three error categories (Disfluent, Off Target, Wrong Context) each with several subtypes.

[4] https://huggingface.co/facebook/bart-large-cnn

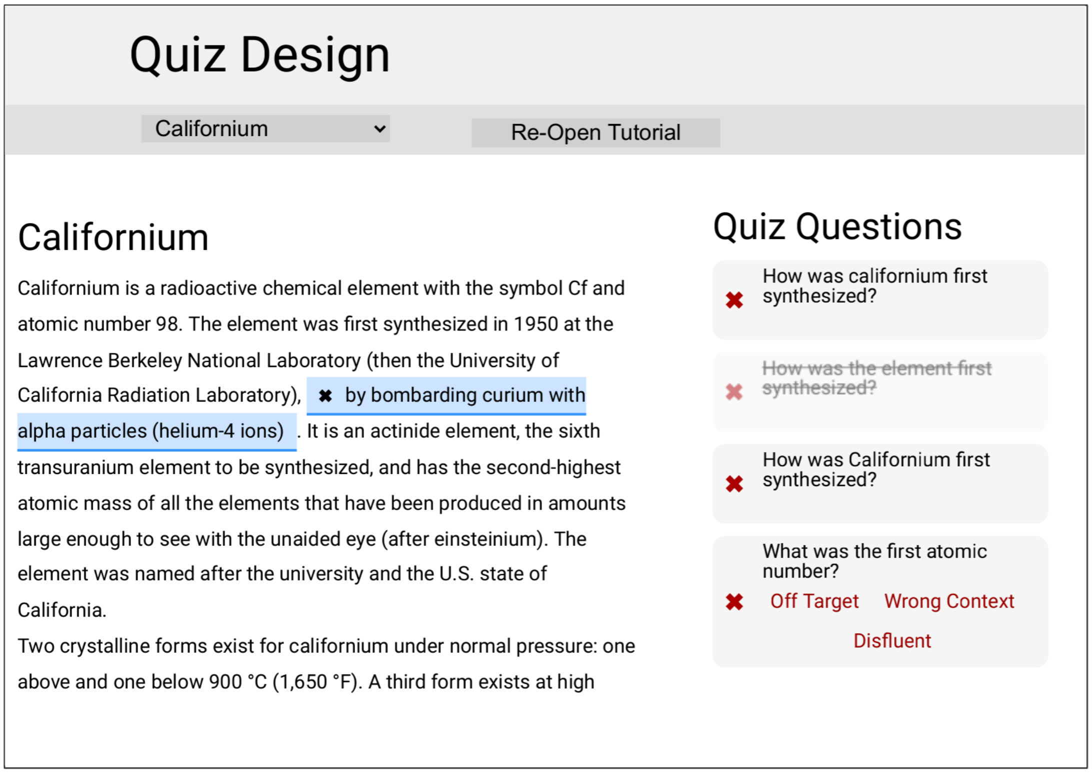

Figure 4: **Screenshot of annotation interface used for the Quiz Design Task.** The teacher has selected the concept highlighted in blue in the reading material in the left column. In the right column, the system gives proposes candidate questions, which can be added to the quiz, or refused with a reason.

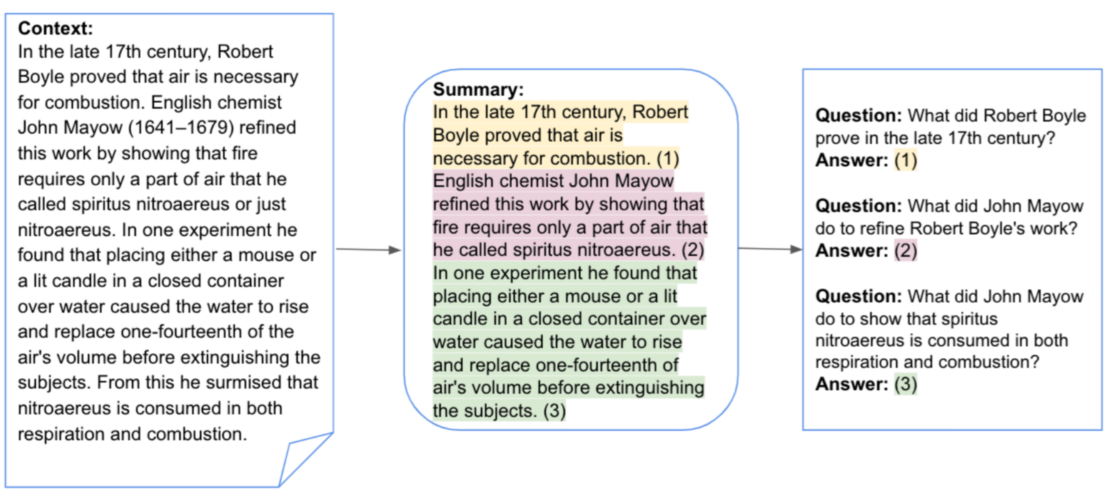

Figure 5: Example of generating QA pairs using summarization and MixQG.

Dataset	Source	Questions
SQuAD	Gold	What happened to NASA's yearly budget after the first landing?
	MixQG	What happened to NASA's budget after the first successful moon landing?
	Gold	How many of Warsaw's inhabitants spoke Polish in 1933?
	MixQG	In 1933, how many of Warsaw's inhabitants were of Polish mother tongue?
	Gold	How long does it take for new areas to have significant oil production?
	MixQG	How long did it take to develop new oil fields?
NarQA	Gold	What are Mulder and Scully doing at the beginning of the story?
	MixQG	What are Mulder and Scully doing in Dallas?
	Gold	How does Chris make extra money?
	MixQG	What does Chris Hughes do for a living?
	Gold	Who died in this story?
	MixQG	What is the name of Mortimer Tregennis's sister?
MCTest	Gold	How many of Mikes friends came to the party
	MixQG	How many of Mike's friends came to the party?
	Gold	Where did Jenny want to go on a trip to?
	MixQG	Where did Jenny want to go on a trip?
	Gold	What was Stewart doing at the beach?
	MixQG	What was Stewart doing when he saw the bug?
BoolQ	Gold	is there an amtrak station in pensacola florida
	MixQG	is there an amtrak station in pensacola florida
	Gold	can a minor sit at a bar in nj
	MixQG	can a minor drink at a bar in new jersey
	Gold	can a polar bear and a grizzly mate
	MixQG	is there such a thing as a grizzly polar bear
QAConv	Gold	What Xbox 360 game is the best dance game for a wife on the Wii?
	MixQG	What is a good dance game for the Wii?
	Gold	Which state is taking its proper place and represents the country's diversity much better?
	MixQG	What state is taking its proper place as early in the primary?
	Gold	In what month will the BLM say go ahead and start the stimulation experiment ?
	MixQG	When will the experiment start?

Table 5: Example of MixQG-3B generated questions and their corresponding ground-truth labels.

Context	**Model**	**Question**	**Annotation**
For example, californium can be used [...] when studying materials using **neutron diffraction and neutron spectroscopy**.	MixQG-L	What methods are used to study materials using californium as a source of neutrons?	No Error
	BART-L	What is Californium used to study materials?	Disfluent
With the exception of some ectothermic species [...] no tetrapods **weighing more than 25 kilograms** (55 pounds) survived.	MixQG-L	What size tetrapods did not survive the extinction?	No Error
	ProphetNet	How much did tetrapods weigh at the time of the Cretaceous-Paleogene extinction?	Off Tgt.
The two DNA strands are known as polynucleotides as they are composed of **simpler monomeric units called nucleotides**.	MixQG-L	What are polynucleotides composed of?	No Error
	BART-L	What are polynucleotides?	Off Tgt.
The Statue of Liberty (**Liberty Enlightening the World**) is a colossal neoclassical sculpture on [...]	ProphetNet	What is another name for the Statue of Liberty?	No Error
	MixQG-L	What is the English translation of the Statue of Liberty?	Off Tgt.
Californium. The element was named **after the university and the U.S. state of California**.	ProphetNet	What is Californium named after?	No Error
	MixQG-L	Where did Californium get its name?	Wrong Ctxt
Fossil fuels provide 85% of the world's energy consumption and the energy system [...]	BART-L	How much of the world's energy consumption does fossil fuels provide?	No Error
	MixQG-L	What percentage of the world's energy consumption is fossil fuels?	Disfluent

Table 6: **Success and failure cases of the MixQG model from the Quiz Design evaluation.** Comparisons to the ProphetNet and BART-Large models are included, with each model receiving the context with a target answer (in bold), and being annotated with an error label by a teacher.

Model	BLEU	R1	R2	RL	RLsum	METEOR	BERTScore
ProphetNet$_{large}$	22.88	51.37	29.48	47.11	47.09	41.46	0.4931
MixQG$_{large}^{BART}$	23.30	54.44	31.92	50.18	50.18	43.47	0.5622
MixQG$_{base}$	23.53	54.39	32.06	50.05	50.02	43.83	0.5566
MixQG$_{large}$	24.42	55.52	33.13	50.99	50.97	45.07	0.5699
MixQG$_{3b}$	**25.42**	**56.11**	**33.91**	**51.85**	**51.86**	**45.75**	**0.5789**

Table 7: Evaluation of differently-sized MixQG models on SQuAD. Base, Large and 3B refer to model configurations with 220 million, 770 million and 3 billion parameters, respectively.