Open-World Factually Consistent Question Generation

Himanshu Maheshwari, Sumit Shekhar, Apoorv Saxena, Niyati Chhaya
Adobe Research, India
{himahesh, sushekha, apoorvs, nchhaya} @ adobe.com

Abstract

Question generation methods based on pre-trained language models often suffer from factual inconsistencies and incorrect entities and are not answerable from the input paragraph. Domain shift – where the test data is from a different domain than the training data - further exacerbates the problem of hallucination. This is a critical issue for any natural language application doing question generation. In this work, we propose an effective data processing technique based on de-lexicalization for consistent question generation across domains. Unlike existing approaches for remedying hallucination, the proposed approach does not filter training data and is generic across question-generation models. Experimental results across six benchmark datasets show that our model is robust to domain shift and produces entity-level factually consistent questions without significant impact on traditional metrics.

1 Introduction

Question generation is the task of generating a question that is relevant to and answerable by a piece of text (Krishna and Iyyer (2019), Chen et al. (2020), Zhu and Hauff (2021), Ushio et al. (2022),). It is an important task in language generation (Fabbri et al. (2020), Yu et al. (2020b)), education (Wang et al. (2022)), and information retrieval (Yu et al. (2020a)). A critical metric for question generation is factual consistency, i.e., the question has facts that are derivable from the input paragraph. This work proposes novel methods to improve entity-level factual consistency while agnostic to model and underlying training data. Nan et al. (2021) and Xiao and Carenini (2022) solve a similar problem for summarization. However, to the best of our knowledge, no work addresses the issue of entity-level factual inconsistency for question generation.

Nema and Khapra (2018) have shown that name entities are essential for a question's answerability. The presence of wrong entities may make the question nonsensical and unanswerable. Table 1 shows entity-level factual inconsistency in question generation by a fine-tuned PEGASUS (Zhang et al., 2019) model. In the first entity, "Kim Jong Un", and in the second example, "Chicago" are hallucinated.

Unlike previous work in the summarization field (Nan et al. (2021), Liu et al. (2021a), Xiao and Carenini (2022)), our work is independent of the model or training process. We also do not reduce dataset size by filtering. Instead, we pre-process datasets to force the model to generate questions faithful to the input using strategies of de-lexicalization and multi-generation and recommend the best strategy. The proposed method improves the factual consistency by $84-100\%$ across multiple datasets while having minimal impact on traditional performance metrics.

We experimented with two popular language models viz. PEGASUS-large and BART-large (Lewis et al., 2020). Our proposed approach consistently performs better for both the language models than normal finetuning. We also compare our approach to recent methods for addressing hallucination in summarization, and our results showed significantly better performance.

2 Related Work

Early work proposed rule-based approaches to generate questions (Labutov et al. (2015)). Recent work using neural models for high-quality generation (Du and Cardie (2017), Du et al. (2017), Wang et al. (2022)). Several works have aimed to control hallucination in different NLP tasks like summarization (Cao et al. (2018), Song et al. (2020), Zhu et al. (2021)), Wu et al. (2021)), generative question answering (Bi et al. (2019), Fan et al. (2019a), Su et al. (2022)), data to text (Nie et al. (2019), Wang et al. (2020), Wang et al. (2021)), and machine translation (Xu et al. (2019), Wang and Sennrich (2020), Zhou et al. (2021)).

Findings of the Association for Computational Linguistics: ACL 2023, pages 2390–2404
July 9-14, 2023

Input: As the president of South Korea, she turned out to be heavily influenced by a cult and giving them access to political documents and basically letting that cult run the country through her. One of her "speech-writers" was a religious figure in this cult that totally exploited the president, who pretty much only got elected because her father had been president, using her for money and power and wealth. The people of South Korea got upset when they learned that the real power that was running their country was the religious group that their president belonged to.
Normal Finetuning: What did Kim Jong Un do wrong?
Rare word delexicalization + Multiple (Ours): Why did the people of South Korea vote out the president of the South Korea?
Input: As I understand it, there's no established *process* for removing the mayor from his post. If he doesn't resign, there's no mechanism for removing him. Really, I don't understand how this guy hasn't lost his job yet.
Normal Finetuning: Why hasn't the mayor of Chicago been fired yet?
Rare word delexicalization + Multiple (Ours): Why hasn't the mayor been fired yet?

Table 1: Qualitative Examples. For detailed analysis refer to section 6.4.

Previous work has explored entity-based de-lexicalization in settings like adapting parser for a new language (Zeman and Resnik, 2008), valid reasoning chains in multi-hop question answering (Jhamtani and Clark, 2020), and eliminating diachronic biases in fake news detection (Murayama et al., 2021).

3 Methodology

The objective is to generate relevant and entity-level factually consistent questions which generalise across domains. For this, we propose novel de-lexicalization strategies combined with a multi-generation strategy. De-lexicalization involves replacing named entities with a special token or rare words during training/inference and replacing the original word after generation. The model's vocabulary is expanded to account for the special tokens used in the de-lexicalization strategies.

De-lexicalization Strategies During Training

[Name i] Token: This strategy replaces the named entity with a token [Name i], where i represents the order of the first appearance of the entity in the paragraph and in the question.

[Name i] Token with Push: This strategy is similar to the previous one. The difference is that if the question has a named entity that is not present in the input paragraph, we replace it with [Name j], where the j is a random number between 0 and the total number of named entities in the input paragraph. The intuition here is that we are pushing or explicitly asking the model to generate a named entity already present in the input paragraph.

[Multiple i] Token: The previous two strategies treat all the named entities as similar. In contrast, in this approach, the entity is replaced with its corresponding semantic tags, followed by an integer representing its order of appearance in the paragraph followed by the question. A semantic tag specifies if an entity is name, organization, location, cardinal, etc.

[Multiple i] Token with Push and Delete: This approach is similar to *[Name i] Token with Push* approach with multiple entity types. However, if the question consists of a named entity type not present in the paragraph, it is deleted.

Rare Word token: This strategy de-lexicalizes only the questions. Here we replace the named entities in questions that do not occur in the input paragraph with a rare word. A rare word is a word that occurs 2 to 5 times in the entire training corpus. If an entity occurs in the input paragraph, it is left as it is.

Examples showing different de-lexicalization strategies are present in the Appendix.

Entity Replacement: During testing, from the generated questions, the entities are replaced using a dictionary look-up of the special token. We treat a output as hallucinated if the special token has no corresponding named entity.

Multi-generation: Here, we generate multiple questions during inference by selecting the top five beams from the output of the language model and selecting the one that is factually consistent and has the least perplexity. If no questions are consistent, the generation with the least perplexity is chosen.

Dataset	Train	Dev	Test
ELI5	150,000	6,925	10,000
AskEconomics	-	-	10,067
AskLegal	-	-	98
MS Marco	-	-	1,043
Natural Questions	-	-	5,000
SciQ	-	-	884

Table 2: Statistics for different datasets

4 Example of Different De-lexicalization Strategies

Table 3 illustrates different de-lexicalization strategies proposed in the paper. The question contains the named entity "U.S.," which is not present in the

Original
Input: One way would be to allow unlimited deductions of savings and tax withdrawals as income . So if you buy $ 50,000 in bonds in 2017 , you deduct all that from your income . Then you sale those bonds for $ 55,000 in 2018 , you would add that $ 55,000 to your 2018 income and it 's taxed like any other income . The simplest way to implement that would be to eliminate penalities and caps on IRA accounts .Said my whole question , do n't know what else to say .
Question: How can the U.S. tax system be reformed?
[Name i] Token
Input: [Name 0] way would be to allow unlimited deductions of savings and tax withdrawals as income . So if you buy $ [Name 1] in bonds in [Name 2] , you deduct all that from your income . Then you sale those bonds for $ [Name 3] in [Name 4] , you would add that $ [Name 3] to your [Name 4] income and it 's taxed like any other income . The simplest way to implement that would be to eliminate penalities and caps on IRA accounts .Said my whole question , do n't know what else to say .
Question: How can the [Name 5] tax system be reformed?
[Name i] Token with Push
Input: [Name 0] way would be to allow unlimited deductions of savings and tax withdrawals as income . So if you buy $ [Name 1] in bonds in [Name 2] , you deduct all that from your income . Then you sale those bonds for $ [Name 3] in [Name 4] , you would add that $ [Name 3] to your [Name 4] income and it 's taxed like any other income . The simplest way to implement that would be to eliminate penalities and caps on IRA accounts .Said my whole question , do n't know what else to say .
Question: How can the [Name 3] tax system be reformed?
[Multiple i] Token
Input: [CARDINAL 0] way would be to allow unlimited deductions of savings and tax withdrawals as income . So if you buy $ [MONEY 0] in bonds in [DATE 0] , you deduct all that from your income . Then you sale those bonds for $ [MONEY 1] in [DATE 1] , you would add that $ [MONEY 1] to your [DATE 1] income and it 's taxed like any other income . The simplest way to implement that would be to eliminate penalities and caps on IRA accounts .Said my whole question , do n't know what else to say .
Question: How can the [GPE 0] tax system be reformed?
[Multiple i] Token with Push and Delete
Input: [CARDINAL 0] way would be to allow unlimited deductions of savings and tax withdrawals as income . So if you buy $ [MONEY 0] in bonds in [DATE 0] , you deduct all that from your income . Then you sale those bonds for $ [MONEY 1] in [DATE 1] , you would add that $ [MONEY 1] to your [DATE 1] income and it 's taxed like any other income . The simplest way to implement that would be to eliminate penalities and caps on IRA accounts .Said my whole question , do n't know what else to say .
Question: How can the tax system be reformed?
Rare word Token
Input: One way would be to allow unlimited deductions of savings and tax withdrawals as income . So if you buy $ 50,000 in bonds in 2017 , you deduct all that from your income . Then you sale those bonds for $ 55,000 in 2018 , you would add that $ 55,000 to your 2018 income and it 's taxed like any other income . The simplest way to implement that would be to eliminate penalities and caps on IRA accounts .Said my whole question , do n't know what else to say .
Question: How can the aster tax system be reformed?

Table 3: Examples of different de-lexicalization strategies. For details refer to section 4

Approach w/o Multi-Generation	C.S. ↑	R-1 ↑	R-2 ↑	R-L ↑	PPL ↓	P_{ne}	P_{wne} ↓	Recall ↑	Precision ↑	F1 ↑
Fine-tuned PEGASUS	0.6748	29.4926	11.5852	27.5309	86.0441	26.6800	42.8036	0.3779	0.4067	0.3918
[Name i] Token	0.6504	28.7351	11.1658	26.8213	97.1283	18.1800	35.4235	0.2289	0.2885	0.2553
[Name i] Token with Push	0.6544	28.8616	11.2306	27.0018	104.0066	21.0700	17.8927	0.2862	0.3578	0.3180
[Multiple i] Token	0.6523	28.8050	11.1888	26.9392	96.2436	21.7700	35.1860	0.2718	0.3491	0.3056
[Multiple i] Token with Push and Delete	0.6564	28.8258	11.1455	26.9559	97.1164	19.2800	20.2282	0.2962	0.3788	0.3325
Rare Word Token	**0.6773**	**29.7333**	**11.8060**	**27.7603**	85.4832	19.4300	10.1390	**0.4477**	0.5107	0.4771
Approach with Multi-Generation										
Fine-tuned PEGASUS	0.6672	28.9704	10.7617	26.8001	41.7799	23.0500	5.1600	0.3986	0.4368	0.4168
[Name i] Token	0.6444	28.1856	10.2253	26.0171	46.2771	14.8200	2.3200	0.2552	0.3300	0.2878
[Name i] Token with Push	0.6495	28.4518	10.3465	26.2775	44.5343	17.2600	1.4500	0.3084	0.3957	0.3466
[Multiple i] Token	0.6502	28.4184	10.3503	26.2616	45.5624	16.9100	3.0200	0.2977	0.3879	0.3369
[Multiple i] Token with Push and Delete	0.6513	28.5909	10.4508	26.4515	43.1499	15.6600	1.2600	0.3206	0.4137	0.3613
Rare Word Token	0.6691	29.1550	10.8146	26.9616	**40.2198**	18.4300	**0.6700**	**0.4477**	**0.5179**	**0.4802**
Spancopy (Base model: PEGASUS)										
Without global relevance	0.6643	29.2871	11.3873	27.4839	94.9375	23.2300	27.1201	0.3775	0.4343	0.4039
With global relevance	0.6732	27.4178	10.0934	26.3062	93.6223	22.2900	28.5913	0.2466	0.6777	0.3617

Table 4: Results of various approaches on ELI5 dataset for PEGASUS model. C.S.: Cosine Similarity | R-1: Rouge 1 | R-2: Rouge 2 | R-L: Rouge l | PPL: Perplexity. For detailed analysis refer to section 6.4.

input.

In the *[Name i] Token* strategy, we replace all named entities with [Name i]. Do note that name entity, 55,000, and 2018 occur twice. Each occurrence is replaced with the same token, i.e., both occurrence of 55,000 is replaced with [Name 3]. Since the "U.S." does not occur in the input, we replace it with [Name 5]. Contrary to this, in the *[Name i] Token with Push* strategy, we replace the U.S. with [Name 3], thereby pushing the model to be faithful to the source.

In the *[Multiple i] Token* strategy, instead of replacing named entities with a common [Name] token, we replace them with their semantic token. Thus, 55,000 is replaced with [MONEY 1] and so on. Like before, each occurrence is replaced with

the same token. The U.S. is replaced with [GPE 0] as no entity of type GPE occurs in the input. Contrary to this, the *[Multiple i] Token with Push and Delete* strategy deletes the entity "U.S." as no GPE-type entity exists in the input. If there were a GPE entity in input (not necessarily "U.S."), it would have been replaced with [GPE 0].

In the *Rare Word Token* strategy, the input is unchanged. Since the U.S. does not occur in input, it is replaced with a rare word (aster).

5 Datasets

We use the supervised ELI5 dataset (Fan et al., 2019b) for training. To ensure that the data is of high quality, we remove all the samples where the answer is short (having less than 50 words), or the question does not have a question mark.

We use three publicly available datasets for evaluation across different domains, viz. MS Marco (Bajaj et al., 2016), Natural Questions (Kwiatkowski et al., 2019) and SciQ (Welbl et al., 2017). We also scraped r/AskLegal [1], and r/AskEconomics[2] for testing on finance and legal domains. Table 2 shows the statistics of the dataset.

6 Experiment and Analysis

6.1 Implementation Details

We use publicly available checkpoints of the language models and fine-tune them for 100k steps with a batch size of 12 and using the Adam optimizer (Kingma and Ba, 2014). The learning rate is set to 10^{-5}, and the models are tested on the dev set every 10k steps. The best-performing model on the dev set is used. The model training takes approximately 6 hours on an Nvidia A100 40 GB GPU. Following Nan et al. (2021) we use the Spacy library[3] to identify named entities.

6.2 Evaluation Metrics

We evaluate both the quality and factual consistency of the generated question. The quality is reported using Rouge-1, Rouge-2, Rouge-L (Lin, 2004) scores and cosine similarity between embedding (from *all-mpnet-base-v2* sentence transformer model (Reimers and Gurevych, 2019)) of generated questions and ground truth. We use the perplexity value suggested by Liu et al. (2021b), using a GPT-2 (Radford et al., 2019). To evaluate factual consistency, we use two metrics. The first metric quantifies the degree of hallucination with respect to the ground truth question. We use the precision, recall, and F1 score proposed by Nan et al. (2021). More details about the exact implementation are in the appendix or in their paper. The second metric quantifies the degree of hallucination with respect to the input paragraph. This metric measures, out of all the questions that have named entities, what percentage of questions have named entities not present in the input. Let N_{hne} represent the number of generated questions with a named entity, and N_{wne} represent the number of generated questions with a wrong named entity. N_{total} represents the total number of questions. Do note $N_{total} \neq N_{hne}$, as we can have questions with no named entity in them. Then $N_{hne}/N_{total} * 100$ represents the percentage of questions having a named entity (P_{ne}), and $N_{wne}/N_{hne} * 100$ represents the percentage of questions having the wrong named entity (P_{wne}). A system with a low P_{wne} value and a high F1 score reflects the system is not hallucinating. We want a system with high factual consistency without **significantly** affecting the quality of the questions as measured by the proposed metrics.

6.3 Baseline

We compare our results with the Spancopy method proposed by Xiao and Carenini (2022) for the summarization. We test with and without global relevance in Spancopy having PEGASUS as the base language model.

6.4 Results and Analysis

Due to space constraints, we only present results for PEGASUS-large in the main text. Results for BART-large can be found in the appendix.

Table 4 shows the results of the test set of the ELI5 dataset. The results indicate that the rare word de-lexicalization plus multiple generation approach performs much better than other methods. Compared to a normal fine-tuned PEGASUS model, the P_{wne} score decreases by about 98%, implying that the generated questions are faithful to the input text. Similarly, the F1 score increases by approximately 21%, implying that all the generated questions are faithful to ground truth. In contrast, decrements in other metric scores are less than 6.7%. Overall, rare word de-lexicalization plus multiple generation performs the best in terms of factual consistency and is comparable in other metrics.

[1] https://www.reddit.com/r/AskLegal/

[2] https://www.reddit.com/r/AskEconomics/

[3] https://spacy.io/usage/linguistic-features#named-entities

Approach	C.S. ↑	R-1 ↑	R-2 ↑	R-L ↑	PPL ↓	P_{ne}	P_{wne} ↓	Recall ↑	Precision ↑	F1 ↑
Dataset: MS Marco										
Normal Finetuned PEGASUS	0.6844	37.5444	**19.2335**	36.0351	79.8135	30.9684	41.1765	0.3923	0.3097	0.3462
Rare word delexicalization + Multiple	0.6759	36.0823	17.0396	34.2419	**29.6060**	21.3806	**0.6711**	**0.5391**	**0.5085**	**0.5234**
Spancopy without global relevance	**0.6959**	**37.8456**	19.0003	**36.5207**	98.0016	27.5168	29.9652	0.3934	0.3153	0.3501
Dataset: Natural Questions										
Normal Finetuned PEGASUS	0.5230	27.0457	**10.8578**	**25.8031**	100.3024	72.2200	46.5522	0.2253	0.2089	0.2168
Rare word delexicalization + Multiple	0.5181	**27.0811**	10.7338	25.5182	**39.3770**	59.6000	**7.3200**	**0.2739**	**0.2707**	**0.2723**
Spancopy without global relevance	**0.6305**	12.2695	3.9743	11.0423	128.6204	73.3200	68.2488	0.0821	0.5031	0.1412
Dataset: SciQ										
Normal Finetuned PEGASUS	0.5469	18.2400	4.7044	16.4770	101.3655	10.0679	35.9551	0.2292	0.2083	0.2183
Rare word delexicalization + Multiple	0.5346	**20.5115**	4.5767	**17.9713**	**31.9291**	5.4299	**0.1131**	**0.4500**	**0.4500**	**0.4500**
Spancopy without global relevance	**0.5613**	18.8779	**4.9120**	17.0202	140.2532	8.1448	18.0556	0.3400	0.4400	0.3836
Dataset: AskEconomics										
Normal Finetuned PEGASUS	0.6250	**34.3724**	**13.1196**	**32.1552**	149.8675	36.4160	39.6890	0.3642	0.3860	0.3748
Rare word delexicalization + Multiple	**0.6260**	33.5555	12.3312	31.0241	**62.5596**	26.5223	**0.6854**	**0.4555**	**0.4976**	**0.4756**
Spancopy without global relevance	0.6222	27.3520	10.6528	25.3469	86.8229	35.0949	25.2194	0.3775	0.4114	0.3937
Dataset: AskLegal										
Normal Finetuned PEGASUS	**0.5963**	**32.0084**	**9.7201**	**29.2130**	104.9676	29.5918	41.3793	0.4583	0.4000	0.4272
Rare word delexicalization + Multiple	0.5943	29.8136	8.8872	26.8056	**65.7854**	18.3674	**1.0204**	**0.6061**	**0.5818**	**0.5937**
Spancopy without global relevance	0.5936	26.2488	9.2795	23.9778	102.6717	29.5918	27.5862	0.3698	0.4375	0.4008

Table 5: Results of Normal finetuned PEGASUS, Rare word delexicalization + Multiple (proposed) and Spancopy without global relevance. C.S.: Cosine Similarity | R-1: Rouge 1 | R-2: Rouge 2 | R-L: Rouge l | PPL: Perplexity. For detailed analysis refer to section 6.4.

The rare word de-lexicalization with multi-generation approach consistently performs better than all the other approaches for all the datasets. Table 5 compares rare word delexicalization + multiple generation with a normal finetuned PEGASUS and Spancopy without global relevance across different datasets. Detailed results for all the approaches across all the datasets are in the appendix.

From the table, it can be seen that rare word delexicalization with multiple generations solves the issue of entity-level inconsistency without negative impact on different metrics. The model was just trained for the ELI5 dataset and was directly used for other datasets. Domain shift exacerbates the issue of entity hallucination, as shown by the P_{wne} value for a normal fine-tuned PEGASUS model, which is usually higher in the presence of domain shift. Thus, our proposed approach works across domains without re-training.

We see that the P_{ne} value decreases across all the datasets for rare word delexicalization with multiple generations. However, this is not wrong. A question without a named entity can still be a valid question (Nema and Khapra, 2018).

Table 1 shows qualitative examples. In the first example, the fine-tuned PEGASUS produces the entity Kim Jong Un that is unfaithful to the source and is entirely unrelated to South Korea. Chicago is hallucinated in the second example. In both examples, our proposed approach generates meaningful and faithful questions. Our approach produces a question with no named entity in the second example, yet the question is meaningful and faithful to the source. This further reinforces our claim that a question without a named entity can still be valid. More outputs can be found in the appendix.

Our approach performs better than the Spancopy architecture (both with and without global relevance). This shows that simple de-lexicalization with multiple generations is better than sophisticated architecture.

7 Conclusion

In this paper, we study the entity-level factual inconsistency in question generation. Our proposed strategy, rare-word de-lexicalization with multi-generation, improve consistency without significantly affecting traditional metrics across data domains. Extensive experimental results further reinforce our claim.

8 Limitations

The P_{ne} value decreased in all datasets. While this is not problematic for question generation, where the presence of a named entity is not always necessary, it does pose an issue for NLG tasks where the inclusion of named entities is important. In these cases, we recommend using alternative techniques that we have proposed. Additionally, using de-lexicalization and over-generation in our approach leads to a high training and inference time.

References

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

Bin Bi, Chen Wu, Ming Yan, Wei Wang, Jiangnan Xia, and Chenliang Li. 2019. Incorporating external knowledge into machine reading for generative 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 2521–2530, Hong Kong, China. Association for Computational Linguistics.

Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2018. Faithful to the original: Fact-aware neural abstractive summarization. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press.

Yu Chen, Lingfei Wu, and Mohammed J. Zaki. 2020. Reinforcement learning based graph-to-sequence model for natural question generation. In *Proceedings of the 8th International Conference on Learning Representations*.

Xinya Du and Claire Cardie. 2017. Identifying where to focus in reading comprehension for neural question generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2067–2073, Copenhagen, Denmark. Association for Computational Linguistics.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. pages 1342–1352.

Alexander Fabbri, Patrick Ng, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. 2020. Template-based question generation from retrieved sentences for improved unsupervised question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4508–4513, Online. Association for Computational Linguistics.

Angela Fan, Claire Gardent, Chloé Braud, and Antoine Bordes. 2019a. Using local knowledge graph construction to scale Seq2Seq models to multi-document inputs. 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 4186–4196, Hong Kong, China. Association for Computational Linguistics.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019b. 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.

Harsh Jhamtani and Peter Clark. 2020. Learning to explain: Datasets and models for identifying valid reasoning chains in multihop question-answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 137–150, Online. Association for Computational Linguistics.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

Kalpesh Krishna and Mohit Iyyer. 2019. Generating question-answer hierarchies. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2321–2334, Florence, Italy. Association for Computational Linguistics.

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.

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.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

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.

Wei Liu, Huanqin Wu, Wenjing Mu, Zhen Li, Tao Chen, and Dan Nie. 2021a. Co2sum:contrastive learning for factual-consistent abstractive summarization.

Yixin Liu, Graham Neubig, and John Wieting. 2021b. On learning text style transfer with direct rewards. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*,

pages 4262–4273, Online. Association for Computational Linguistics.

Taichi Murayama, Shoko Wakamiya, and Eiji Aramaki. 2021. Mitigation of diachronic bias in fake news detection dataset. In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 182–188, Online. Association for Computational Linguistics.

Feng Nan, Ramesh Nallapati, Zhiguo Wang, Cicero Nogueira dos Santos, Henghui Zhu, Dejiao Zhang, Kathleen McKeown, and Bing Xiang. 2021. Entity-level factual consistency of abstractive text summarization. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2727–2733, Online. Association for Computational Linguistics.

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, Brussels, Belgium. Association for Computational Linguistics.

Feng Nie, Jin-Ge Yao, Jinpeng Wang, Rong Pan, and Chin-Yew Lin. 2019. A simple recipe towards reducing hallucination in neural surface realisation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2673–2679, Florence, Italy. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Kaiqiang Song, Logan Lebanoff, Qipeng Guo, Xipeng Qiu, X. Xue, Chen Li, Dong Yu, and Fei Liu. 2020. Joint parsing and generation for abstractive summarization. In *AAAI*.

Dan Su, Xiaoguang Li, Jindi Zhang, Lifeng Shang, Xin Jiang, Qun Liu, and Pascale Fung. 2022. Read before generate! faithful long form question answering with machine reading. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 744–756, Dublin, Ireland. Association for Computational Linguistics.

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

Chaojun Wang and Rico Sennrich. 2020. On exposure bias, hallucination and domain shift in neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3544–3552, Online. Association for Computational Linguistics.

Peng Wang, Junyang Lin, An Yang, Chang Zhou, Yichang Zhang, Jingren Zhou, and Hongxia Yang. 2021. Sketch and refine: Towards faithful and informative table-to-text generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4831–4843, Online. Association for Computational Linguistics.

Xu Wang, Simin Fan, Jessica Houghton, and Lu Wang. 2022. Towards process-oriented, modular, and versatile question generation that meets educational needs.

Zhenyi Wang, Xiaoyang Wang, Bang An, Dong Yu, and Changyou Chen. 2020. Towards faithful neural table-to-text generation with content-matching constraints. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1072–1086, Online. Association for Computational Linguistics.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. 2017. Crowdsourcing multiple choice science questions. *ArXiv*, abs/1707.06209.

Zeqiu Wu, Michel Galley, Chris Brockett, Yizhe Zhang, Xiang Gao, Chris Quirk, Rik Koncel-Kedziorski, Jianfeng Gao, Hannaneh Hajishirzi, Mari Ostendorf, and Bill Dolan. 2021. A controllable model of grounded response generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35:14085–14093.

Wen Xiao and Giuseppe Carenini. 2022. Entity-based spancopy for abstractive summarization to improve the factual consistency.

Weijia Xu, Xing Niu, and Marine Carpuat. 2019. Differentiable sampling with flexible reference word order for neural machine translation. 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 2047–2053, Minneapolis, Minnesota. Association for Computational Linguistics.

Qian Yu, Lidong Bing, Qiong Zhang, Wai Lam, and Luo Si. 2020a. Review-based question generation with adaptive instance transfer and augmentation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 280–290, Online. Association for Computational Linguistics.

Wenhao Yu, Lingfei Wu, Yu Deng, Ruchi Mahindru, Qingkai Zeng, Sinem Guven, and Meng Jiang. 2020b. A technical question answering system with transfer learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 92–99, Online. Association for Computational Linguistics.

Daniel Zeman and Philip Resnik. 2008. Cross-language parser adaptation between related languages. In *Proceedings of the IJCNLP-08 Workshop on NLP for Less Privileged Languages*.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2019. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization.

Chunting Zhou, Graham Neubig, Jiatao Gu, Mona Diab, Francisco Guzmán, Luke Zettlemoyer, and Marjan Ghazvininejad. 2021. Detecting hallucinated content in conditional neural sequence generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1393–1404, Online. Association for Computational Linguistics.

Chenguang Zhu, William Hinthorn, Ruochen Xu, Qingkai Zeng, Michael Zeng, Xuedong Huang, and Meng Jiang. 2021. Enhancing factual consistency of abstractive summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 718–733, Online. Association for Computational Linguistics.

Peide Zhu and Claudia Hauff. 2021. Evaluating bert-based rewards for question generation with reinforcement learning. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*, ICTIR '21, page 261–270, New York, NY, USA. Association for Computing Machinery.

A Processing Publicly Available Datasets

This section describes our processing for MS Marco, Natural Questions, and SciQ datasets. Since these datasets are used exclusively for testing, we can even use their training set for testing. For MS Marco, we use their train set due to the small size of the test set. Since MS Marco is a sentence-based dataset, we usually see small input contexts. So we only include those data points where the answer has at least 40 words, and the question ends with a question mark. We also use the training set for Natural questions as it is a well-defined JSON file. We randomly select five thousand questions from the training set. We also ensure that the answer is not from the table. We use the test set for the SciQ dataset; however, we filter out all the documents for which supporting text is missing. This supporting text is the input to the model.

B Precision, Recall and F1 Scores

Let q_{gt} and q_{gen} be the ground truth and generated questions, respectively. Let $N(q_{gt} \cap q_{gen})$ represent the number of named entities common between ground truth and generated question. Similarly, N(q_{gt}) and N(q_{gen}) represent the number of names entities in the ground truth and generated question, respectively. Thus, the precision is: $N(q_{gt} \cap q_{gen})/N(q_{gen})$ and recall is: $N(q_{gt} \cap q_{gen})/N(q_{gt})$. The F1 score is the harmonic mean of recall and precision.

C Results Across Multiple Datasets

This section presents the results of different delexicalization strategies across different datasets. Table 6, 7, 8, 9, and 10 present the results for MS Marco, natural questions, SciQ, AskEconomics, and AskLegal datasets for the PEGASUS model. Table 11, 12, 13, 14, 15, and 16 present the results for ELI5, MS Marco, natural questions, SciQ, AskEconomics, and AskLegal datasets for the BART model.

D More Qualitative Examples

Table 17 shows some more qualitative examples.

Approach w/o Multi-Generation	C.S. ↑	R-1 ↑	R-2 ↑	R-L ↑	PPL ↓	P_{ne}	P_{wne} ↓	Recall ↑	Precision ↑	F1 ↑
Fine-tuned PEGASUS	0.6844	37.5444	19.2335	36.0351	79.8135	30.9684	41.1765	0.3923	0.3097	0.3462
[Name i] Token	0.6361	36.1754	18.1748	34.9726	93.6765	23.5858	35.3659	0.2347	0.2303	0.2324
[Name i] Token with Push	0.6466	36.0127	18.3969	34.8516	94.1234	27.0374	15.9574	0.2663	0.2732	0.2697
[Multiple i] Token	0.6385	35.8072	18.1312	34.6703	90.9868	29.3384	32.0261	0.3035	0.3158	0.3095
[Multiple i] Token with Push and Delete	0.6530	36.0869	17.9565	34.8825	90.9744	24.4487	18.0392	0.3406	0.3261	0.3332
Rare Word Token	0.6903	38.0898	**19.6516**	**36.6146**	74.4608	23.1064	20.3320	0.4783	0.4237	0.4493
Approach with Multi-Generation										
Fine-tuned PEGASUS	0.6725	35.3435	16.9061	33.7758	34.8961	26.6539	5.0815	0.4354	0.3874	0.4100
[Name i] Token	0.6301	34.3547	16.0744	32.8485	35.6354	17.8332	1.9175	0.1936	0.2258	0.2084
[Name i] Token with Push	0.6311	34.4746	16.2056	33.0859	34.7968	21.2848	0.7670	0.2922	0.2963	0.2942
[Multiple i] Token	0.6293	34.4290	16.3977	33.0232	37.1887	22.9147	3.7392	0.3333	0.3546	0.3437
[Multiple i] Token with Push and Delete	0.6377	34.5643	16.4035	33.2079	32.2173	19.4631	1.0546	0.3862	0.4024	0.3941
Rare Word Token	0.6759	36.0823	17.0396	34.2419	**29.6060**	21.3806	**0.6711**	**0.5391**	**0.5085**	**0.5234**
Spancopy (Base model: PEGASUS)										
Without global relevance	0.6959	**37.8456**	19.0003	36.5207	98.0016	27.5168	29.9652	0.3934	0.3153	0.3501
With global relevance	**0.6961**	37.8070	18.8587	36.4055	93.6907	27.4209	26.9231	0.3836	0.3506	0.3664

Table 6: Results of various approaches on MS Marco dataset for PEGASUS model. C.S.: Cosine Similarity | R-1: Rouge 1 | R-2: Rouge 2 | R-L: Rouge l | PPL: Perplexity.

Approach w/o Multi-Generation	C.S. ↑	R-1 ↑	R-2 ↑	R-L ↑	PPL ↓	P_{ne}	P_{wne} ↓	Recall ↑	Precision ↑	F1 ↑
Fine-tuned PEGASUS	0.5230	27.0457	10.8578	25.8031	100.3024	72.2200	46.5522	0.2253	0.2089	0.2168
[Name i] Token	0.3936	22.9647	8.1260	21.9904	171.2617	60.9400	40.3676	0.0825	0.0911	0.0866
[Name i] Token with Push	0.4290	24.1552	8.8400	23.1948	208.5758	62.7400	16.4170	0.1362	0.1505	0.1430
[Multiple i] Token	0.4103	23.3620	8.4394	22.3829	180.8732	69.7400	37.4821	0.1547	0.1766	0.1650
[Multiple i] Token with Push and Delete	0.4146	23.4967	8.4898	22.5391	200.1246	54.0400	17.6906	0.1779	0.2050	0.1905
Rare Word Token	0.5249	**27.5742**	**11.3235**	**26.4109**	98.1137	63.4400	25.9142	0.2507	0.2483	0.2495
Approach with Multi-Generation										
Fine-tuned PEGASUS	0.5201	26.8340	10.5054	25.2617	51.1872	66.5600	19.3600	0.2587	0.2418	0.2500
[Name i] Token	0.3938	23.0861	7.9811	21.9169	73.3829	54.7000	11.6800	0.0915	0.1070	0.0987
[Name i] Token with Push	0.4201	24.0322	8.6477	22.8173	64.4276	57.3400	**2.6800**	0.1316	0.1512	0.1407
[Multiple i] Token	0.4199	23.6249	8.3804	22.4047	65.6907	62.5600	11.3000	0.1558	0.1783	0.1663
[Multiple i] Token with Push and Delete	0.4111	23.4675	8.1350	22.2492	49.5079	47.8800	2.7400	0.1805	0.2120	0.1950
Rare Word Token	0.5181	27.0811	10.7338	25.5182	**39.3770**	59.6000	7.3200	**0.2739**	**0.2707**	**0.2723**
Spancopy (Base model: PEGASUS)										
Without global relevance	**0.6305**	12.2695	3.9743	11.0423	128.6204	73.3200	68.2488	0.0821	0.5031	0.1412
With global relevance	0.6268	12.2730	4.1031	11.1419	117.6533	69.4000	66.6859	0.0755	0.4763	0.1303

Table 7: Results of various approaches on Natural Questions dataset for PEGASUS model. C.S.: Cosine Similarity | R-1: Rouge 1 | R-2: Rouge 2 | R-L: Rouge l | PPL: Perplexity.

Approach w/o Multi-Generation	C.S. ↑	R-1 ↑	R-2 ↑	R-L ↑	PPL ↓	P_{ne}	P_{wne} ↓	Recall ↑	Precision ↑	F1 ↑
Fine-tuned PEGASUS	0.5469	18.2400	4.7044	16.4770	101.3655	10.0679	35.9551	0.2292	0.2083	0.2183
[Name i] Token	0.5419	18.7286	4.4430	16.8223	102.1005	6.7873	38.3333	0.2000	0.2000	0.2000
[Name i] Token with Push	0.5361	18.2758	4.4104	16.4034	106.3667	8.8235	19.2308	0.2750	0.3000	0.2870
[Multiple i] Token	0.5375	18.5645	4.2903	16.4698	96.3949	8.5973	38.1579	0.4063	0.4375	0.4213
[Multiple i] Token with Push and Delete	0.5431	18.5072	4.2430	16.4152	94.8746	7.2398	23.4375	0.3889	0.4444	0.4148
Rare Word Token	0.5502	18.6446	4.6170	16.5633	108.7372	6.5611	8.6207	0.4318	0.4394	0.4356
Approach with Multi-Generation										
Fine-tuned PEGASUS	0.5375	20.4049	4.7434	**18.0692**	34.7327	7.9186	1.1312	0.4318	0.4318	0.4318
[Name i] Token	0.5237	19.7210	4.2975	17.5061	31.5745	4.8643	0.5656	0.2308	0.2308	0.2308
[Name i] Token with Push	0.5264	19.9181	4.3193	17.5872	32.4990	6.4480	0.3394	**0.4500**	**0.4500**	**0.4500**
[Multiple i] Token	0.5212	19.7287	4.0927	17.3978	**30.7673**	5.9955	0.7919	0.4375	0.5000	0.4667
[Multiple i] Token with Push and Delete	0.5254	19.5619	4.1523	17.1332	31.3228	5.0905	0.2262	0.3947	0.3947	0.3947
Rare Word Token	0.5346	**20.5115**	4.5767	17.9713	31.9291	5.4299	**0.1131**	**0.4500**	**0.4500**	**0.4500**
Spancopy (Base model: PEGASUS)										
Without global relevance	**0.5613**	18.8779	**4.9120**	17.0202	140.2532	8.1448	18.0556	0.3400	0.4400	0.3836
With global relevance	0.5566	18.2197	4.5123	16.4520	128.5693	7.6923	19.1177	0.3636	0.4091	0.3850

Table 8: Results of various approaches on SciQ dataset for PEGASUS model. C.S.: Cosine Similarity | R-1: Rouge 1 | R-2: Rouge 2 | R-L: Rouge l | PPL: Perplexity.

Approach w/o Multi-Generation	C.S. ↑	R-1 ↑	R-2 ↑	R-L ↑	PPL ↓	P_{ne}	P_{wne} ↓	Recall ↑	Precision ↑	F1 ↑
Fine-tuned PEGASUS	0.6250	34.3724	13.1196	32.1552	149.8675	36.4160	39.6890	0.3642	0.3860	0.3748
[Name i] Token	0.6038	34.6158	12.7742	32.4235	161.2746	24.5554	34.2638	0.2237	0.2750	0.2467
[Name i] Token with Push	0.6117	**35.4423**	13.3007	**33.1954**	169.9248	29.9394	17.3855	0.3362	0.3884	0.3604
[Multiple i] Token	0.6028	34.6618	12.8921	32.4284	158.3679	37.3994	49.9602	0.3152	0.3844	0.3464
[Multiple i] Token with Push and Delete	0.6123	34.9912	13.1035	32.7917	157.6627	25.9859	16.6284	0.3465	0.4084	0.3749
Rare Word Token	**0.6303**	35.2248	**13.8321**	32.9506	149.4177	27.5057	8.0173	**0.4581**	**0.5018**	**0.4790**
Approach with Multi-Generation										
Fine-tuned PEGASUS	0.6201	32.7750	11.7094	30.3072	69.9868	33.2174	6.8243	0.3979	0.4257	0.4113
[Name i] Token	0.6051	33.3825	11.8935	30.9546	73.2065	21.2178	3.1290	0.2801	0.3316	0.3037
[Name i] Token with Push	0.6085	33.8649	12.1014	31.4369	71.4164	26.0058	2.1158	0.3501	0.4080	0.3768
[Multiple i] Token	0.6048	33.5993	11.8591	31.2375	72.2068	25.1614	4.4999	0.3366	0.4072	0.3685
[Multiple i] Token with Push and Delete	0.6084	33.6636	11.9805	31.3073	68.3048	22.7476	1.3410	0.3631	0.4354	0.3960
Rare Word Token	0.6260	33.5555	12.3312	31.0241	**62.5596**	26.5223	**0.6854**	0.4555	0.4976	0.4756
Spancopy (Base model: PEGASUS)										
Without global relevance	0.6222	27.3520	10.6528	25.3469	86.8229	35.0949	25.2194	0.3775	0.4114	0.3937
With global relevance	0.6234	27.3804	10.7121	25.3495	93.4049	33.5651	26.0728	0.2855	0.4279	0.4056

Table 9: Results of various approaches on AskEconomics dataset for PEGASUS model. C.S.: Cosine Similarity | R-1: Rouge 1 | R-2: Rouge 2 | R-L: Rouge l | PPL: Perplexity.

Approach w/o Multi-Generation	C.S. ↑	R-1 ↑	R-2 ↑	R-L ↑	PPL ↓	P_{ne}	P_{wne} ↓	Recall ↑	Precision ↑	F1 ↑
Fine-tuned PEGASUS	0.5963	32.0084	9.7201	29.2130	104.9676	29.5918	41.3793	0.4583	0.4000	0.4272
[Name i] Token	0.5918	32.7445	10.3777	30.1671	156.3077	17.3469	35.2941	0.4242	0.4546	0.4389
[Name i] Token with Push	0.5806	31.7500	10.1226	28.9167	157.8613	20.4083	30.0000	0.2424	0.1591	0.1921
[Multiple i] Token	0.5924	31.6534	9.6734	28.7415	143.8824	23.4694	34.7826	0.1852	0.2593	0.2593
[Multiple i] Token with Push and Delete	0.5832	31.6748	10.2476	29.0825	129.0326	16.3265	25.0000	0.3667	0.3833	0.3748
Rare Word Token	**0.6073**	**34.1803**	**12.1201**	**31.3956**	150.7832	22.4490	9.0909	0.5333	0.4933	0.5126
Approach with Multi-Generation										
Fine-tuned PEGASUS	0.5945	31.0129	8.8776	28.1593	65.7903	25.5102	7.1429	0.4583	0.4000	0.4272
[Name i] Token	0.5812	30.4870	8.7033	28.0827	62.2575	9.1837	2.0408	0.3333	0.3333	0.3333
[Name i] Token with Push	0.5733	29.7218	8.3545	26.6581	65.7451	19.3878	3.0612	0.3333	0.3939	0.3611
[Multiple i] Token	0.5759	29.9117	7.9798	26.7829	**56.3279**	18.3674	2.0408	0.3333	0.3485	0.3407
[Multiple i] Token with Push and Delete	0.5683	30.7385	8.3835	27.6193	56.9851	18.3674	**1.0204**	0.3205	0.3462	0.3328
Rare Word Token	0.5943	29.8136	8.8872	26.8056	65.7854	18.3674	**1.0204**	**0.6061**	**0.5818**	**0.5937**
Spancopy (Base model: PEGASUS)										
Without global relevance	0.5936	26.2488	9.2795	23.9778	102.6717	29.5918	27.5862	0.3698	0.4375	0.4008
With global relevance	0.5992	26.5635	9.4533	24.2483	107.6989	23.4694	39.1304	0.3889	0.4167	0.4023

Table 10: Results of various approaches on AskLegal dataset for PEGASUS model. C.S.: Cosine Similarity | R-1: Rouge 1 | R-2: Rouge 2 | R-L: Rouge l | PPL: Perplexity.

Approach w/o Multi-Generation	C.S. ↑	R-1 ↑	R-2 ↑	R-L ↑	PPL ↓	P_{ne}	P_{wne} ↓	Recall ↑	Precision ↑	F1 ↑
Fine-tuned BART	0.6708	**30.3458**	**12.4110**	**28.4024**	84.2910	23.8800	26.0888	0.4112	0.4634	0.4358
[Name i] Token	0.6392	29.2884	11.5541	27.4596	104.4070	19.6000	77.4490	0.0670	0.0868	0.0756
[Name i] Token with Push	0.6566	29.6925	11.7600	27.8134	108.5305	20.5900	19.7669	0.2475	0.3175	0.2782
[Multiple i] Token	0.6601	30.2565	10.8062	27.8040	94.4202	19.5700	20.3884	0.3254	0.4031	0.3601
[Multiple i] Token with Push and Delete	0.6681	30.2860	11.9954	28.2523	83.4336	18.4200	18.0239	0.3339	0.4126	0.3691
Rare Word Token	**0.6723**	30.3140	12.2587	28.3881	85.8617	19.7900	9.6513	0.4320	0.5073	0.4667
Approach with Multi-Generation										
Fine-tuned BART	0.6619	29.6183	11.2371	27.3785	40.7329	21.7400	2.7700	0.4323	0.4972	0.4625
[Name i] Token	0.6409	28.8991	10.6597	26.7667	52.8637	14.8600	8.4600	0.1294	0.1661	0.1455
[Name i] Token with Push	0.6501	29.3093	10.9578	27.0919	46.9002	17.5900	1.9400	0.2656	0.3409	0.2986
[Multiple i] Token	0.6543	29.5429	10.9538	27.1540	42.1324	16.2200	1.4800	0.3400	0.4302	0.3798
[Multiple i] Token with Push and Delete	0.6592	29.5740	10.8002	27.1938	**39.1928**	16.0800	1.2000	0.3499	0.4423	0.3907
Rare Word Token	0.6624	29.5117	11.0852	27.2535	39.3360	18.8700	**0.5800**	**0.4396**	**0.5149**	**0.4743**

Table 11: Results of various approaches on ELI5 dataset for BART model. C.S.: Cosine Similarity | R-1: Rouge 1 | R-2: Rouge 2 | R-L: Rouge l | PPL: Perplexity.

Approach w/o Multi-Generation	C.S. ↑	R-1 ↑	R-2 ↑	R-L ↑	PPL ↓	P_{ne}	P_{wne} ↓	Recall ↑	Precision ↑	F1 ↑
Fine-tuned BART	**0.6975**	38.6659	19.6163	37.1822	80.2672	30.0096	28.4345	0.4534	0.3757	0.4109
[Name i] Token	0.6255	36.6308	18.5853	35.5317	103.4031	27.5168	77.7004	0.0264	0.0242	0.0253
[Name i] Token with Push	0.6738	37.3418	19.1817	36.2664	109.8183	29.0508	13.2013	0.2408	0.2209	0.2304
[Multiple i] Token	0.6731	37.6786	18.9069	36.4120	87.9487	24.3528	21.2598	0.2921	0.2832	0.2876
[Multiple i] Token with Push and Delete	0.6721	36.8989	18.3516	35.5843	72.8703	24.9281	23.4615	0.3185	0.2937	0.3056
Rare Word Token	0.6957	**38.6819**	**19.6355**	**37.2470**	84.3463	25.0240	16.0920	**0.5015**	**0.4688**	**0.4846**
Approach with Multi-Generation										
Fine-tuned BART	0.6818	37.1152	18.0769	35.2690	33.8383	26.7498	2.8763	0.4505	0.4189	0.4341
[Name i] Token	0.6267	35.0016	16.4603	33.5250	52.8990	20.5177	12.5599	0.0381	0.0458	0.0416
[Name i] Token with Push	0.6607	35.8644	17.0981	34.4245	39.8618	22.7229	**0.5753**	0.3035	0.3158	0.3095
[Multiple i] Token	0.6551	35.5674	16.6391	33.9338	31.5476	21.7641	1.3423	0.2943	0.2926	0.2934
[Multiple i] Token with Push and Delete	0.6444	34.4571	15.9317	32.7743	**29.9093**	20.6136	1.5340	0.3261	0.3025	0.3139
Rare Word Token	0.6814	36.9171	17.7233	34.9347	30.3011	23.6817	1.1505	0.4877	0.4568	0.4717

Table 12: Results of various approaches on MS Marco dataset for BART model. C.S.: Cosine Similarity | R-1: Rouge 1 | R-2: Rouge 2 | R-L: Rouge l | PPL: Perplexity.

Approach w/o Multi-Generation	C.S. ↑	R-1 ↑	R-2 ↑	R-L ↑	PPL ↓	P_{ne}	P_{wne} ↓	Recall ↑	Precision ↑	F1 ↑
Fine-tuned BART	0.5355	**28.2737**	**11.7518**	**26.9399**	91.7022	71.5200	29.6421	0.2369	0.2159	0.2260
[Name i] Token	0.3888	23.1978	8.3257	22.4231	188.8991	66.5400	63.8714	0.0310	0.0363	0.0334
[Name i] Token with Push	0.4731	26.0937	10.2157	25.1520	243.9011	67.7600	12.3672	0.1571	0.1697	0.1631
[Multiple i] Token	0.4584	25.6700	10.0286	24.6388	200.9676	68.3800	24.7442	0.1750	0.1875	0.1810
[Multiple i] Token with Push and Delete	0.4740	26.7674	10.5108	25.5187	155.8149	60.3600	20.9742	0.1759	0.1821	0.1789
Rare Word Token	**0.5358**	27.9430	11.3583	26.5285	88.2666	67.3800	18.7296	0.2477	0.2307	0.2389
Approach with Multi-Generation										
Fine-tuned BART	0.5277	27.9300	11.3603	26.1017	42.0300	68.3600	11.1600	0.2413	0.2222	0.2314
[Name i] Token	0.4092	24.0923	8.7261	22.8177	114.3201	57.6600	24.1600	0.0486	0.0579	0.0528
[Name i] Token with Push	0.4641	25.6506	9.6886	24.1816	65.0078	61.8800	**2.2800**	0.1621	0.1771	0.1693
[Multiple i] Token	0.4589	25.8030	9.8690	24.2661	54.9045	62.0000	7.0000	0.1798	0.1867	0.1832
[Multiple i] Token with Push and Delete	0.4607	26.1420	9.8590	24.5867	47.0870	53.7200	4.6200	0.1786	0.1902	0.1842
Rare Word Token	0.5263	27.5103	10.8879	25.6209	**37.3464**	63.8000	5.3200	**0.2539**	**0.2419**	**0.2477**

Table 13: Results of various approaches on Natural Questions dataset for BART model. C.S.: Cosine Similarity | R-1: Rouge 1 | R-2: Rouge 2 | R-L: Rouge l | PPL: Perplexity.

Approach w/o Multi-Generation	C.S. ↑	R-1 ↑	R-2 ↑	R-L ↑	PPL ↓	P_{ne}	P_{wne} ↓	Recall ↑	Precision ↑	F1 ↑
Fine-tuned BART	0.5435	19.0690	5.5486	17.2327	94.8594	8.2579	16.4384	0.5375	0.5250	0.5312
[Name i] Token	0.5529	18.8496	5.3620	16.9887	122.1728	5.6561	68.0000	0.0513	0.0513	0.0513
[Name i] Token with Push	**0.5561**	19.6120	**5.8026**	17.7135	124.1907	7.9186	15.7143	0.4423	0.4423	0.4423
[Multiple i] Token	0.5473	19.4473	5.3066	17.3885	100.0570	5.5430	6.1225	0.4000	0.3500	0.3733
[Multiple i] Token with Push and Delete	0.5455	19.4403	5.5812	17.2885	84.3905	4.2986	13.1579	0.4615	0.4615	0.4615
Rare Word Token	0.5405	19.3497	5.6073	17.3604	100.1989	5.8823	9.6154	0.5588	0.5588	0.5588
Approach with Multi-Generation										
Fine-tuned BART	0.5281	20.5964	4.6589	18.0049	29.1067	7.2398	0.1131	0.5000	0.5000	0.5000
[Name i] Token	0.5386	21.1234	5.2541	18.5160	37.9603	3.3937	0.6787	0.3571	0.3571	0.3571
[Name i] Token with Push	0.5401	**21.4787**	5.4889	**18.7693**	32.2501	4.0724	0.2262	0.5000	0.5625	0.5294
[Multiple i] Token	0.5267	20.6369	4.4584	17.7864	28.1237	3.3937	0.1131	0.6667	0.6667	0.6667
[Multiple i] Token with Push and Delete	0.5267	21.1890	4.9835	18.6015	27.5412	3.0543	**0.0000**	**0.8333**	**0.8889**	**0.8602**
Rare Word Token	0.5222	20.6120	4.7786	17.8947	**27.3428**	4.7511	**0.0000**	0.6389	0.6111	0.6247

Table 14: Results of various approaches on SciQ dataset for BART model. C.S.: Cosine Similarity | R-1: Rouge 1 | R-2: Rouge 2 | R-L: Rouge l | PPL: Perplexity.

Approach w/o Multi-Generation	C.S. ↑	R-1 ↑	R-2 ↑	R-L ↑	PPL ↓	P_{ne}	P_{wne} ↓	Recall ↑	Precision ↑	F1 ↑
Fine-tuned BART	0.6258	34.7086	**14.5584**	**32.5739**	153.0107	34.4095	28.1467	0.3817	0.4239	0.4017
[Name i] Token	0.5948	33.9577	13.4035	31.9209	178.6685	28.4097	75.2797	0.0733	0.0830	0.0779
[Name i] Token with Push	0.6144	**34.8292**	14.0905	32.6011	176.4246	30.3665	18.8747	0.2566	0.3085	0.2802
[Multiple i] Token	0.6135	34.4133	13.9942	32.0568	147.5120	28.7772	20.4004	0.3750	0.4273	0.3994
[Multiple i] Token with Push and Delete	0.6223	34.3360	13.8667	32.0147	136.4393	27.0587	15.6755	0.3581	0.4171	0.3853
Rare Word Token	**0.6306**	34.5594	14.4776	32.4075	141.2237	28.3799	8.0854	0.4410	0.4890	0.4638
Approach with Multi-Generation										
Fine-tuned BART	0.6200	32.5616	12.9770	30.2733	73.1045	32.6910	4.5793	0.4052	0.4505	0.4267
[Name i] Token	0.5995	32.0092	12.1092	29.8257	87.9262	22.0622	11.9201	0.1520	0.1797	0.1647
[Name i] Token with Push	0.6117	32.2095	12.5313	29.8737	75.1088	26.9892	2.8606	0.2866	0.3486	0.3146
[Multiple i] Token	0.6118	32.6829	12.7997	30.1692	67.0476	25.8170	2.1357	0.3791	0.4355	0.4053
[Multiple i] Token with Push and Delete	0.6188	32.2967	12.6219	29.8401	**60.3746**	24.3667	1.6291	0.3750	0.4314	0.4012
Rare Word Token	0.6243	31.6490	12.7780	29.3689	61.2904	28.2209	**0.6854**	**0.4486**	**0.4985**	**0.4722**

Table 15: Results of various approaches on AskEconomics dataset for BART model. C.S.: Cosine Similarity | R-1: Rouge 1 | R-2: Rouge 2 | R-L: Rouge l | PPL: Perplexity.

Approach w/o Multi-Generation	C.S. ↑	R-1 ↑	R-2 ↑	R-L ↑	PPL ↓	P_{ne}	P_{wne} ↓	Recall ↑	Precision ↑	F1 ↑
Fine-tuned BART	**0.5747**	31.4161	11.3146	29.2014	153.9654	25.5102	36.0000	0.3944	0.4222	0.4079
[Name i] Token	0.5577	30.6848	11.8267	29.1010	134.7211	19.3878	78.9474	0.0556	0.0556	0.0556
[Name i] Token with Push	0.5455	29.7544	9.9516	27.4427	155.7539	21.4286	23.8095	0.1218	0.2308	0.1594
[Multiple i] Token	0.5639	30.2565	10.8062	27.8040	115.1768	18.3674	22.2222	0.2250	0.3000	0.2571
[Multiple i] Token with Push and Delete	0.5701	29.7627	9.8476	27.0681	136.1808	20.4082	15.0000	0.3056	0.2639	0.2832
Rare Word Token	0.5719	**32.3624**	**12.1716**	**30.3172**	147.6951	22.4490	13.6364	**0.5944**	**0.6222**	**0.6080**
Approach with Multi-Generation										
Fine-tuned BART	0.5517	27.8485	9.2759	25.6313	119.2233	26.5306	3.0612	0.4444	0.3333	0.3810
[Name i] Token	0.5400	28.7264	9.3187	26.6250	**67.6563**	13.2653	6.1224	0.1000	0.1000	0.1000
[Name i] Token with Push	0.5446	28.7321	10.2599	26.3664	82.5094	18.3674	3.0612	0.4524	0.4643	0.4583
[Multiple i] Token	0.5655	27.5241	9.0784	25.0620	94.9724	18.3674	1.0204	0.3611	0.3889	0.3745
[Multiple i] Token with Push and Delete	0.5598	26.7347	9.7592	24.6835	76.4061	16.3265	**0.0000**	0.3250	0.4000	0.3586
Rare Word Token	0.5548	26.5433	8.6191	24.6013	69.1614	21.4286	1.0204	0.5321	0.5385	0.5352

Table 16: Results of various approaches on AskLegal dataset for BART model. C.S.: Cosine Similarity | R-1: Rouge 1 | R-2: Rouge 2 | R-L: Rouge l | PPL: Perplexity.

Input: Because there is not political profit in it. No one gets elected campaigned to strike down laws about where you can tie your horse. It is hard to enough getting relevant laws pass, now one wants to take the effort to repeal laws that aren't really hurting anyone. Especially when they get no credit for doing so. Examples would be like those found in this article _URL_0_

Normal Finetuning: Why are there so many laws that have been struck down by the Supreme Court?

Rare word delexicalization + Multiple: Why is it so hard to pass laws that are actually beneficial to the public?

Input: Theres a treaty forbidding any state from claiming extraterrestrial land. That being said, Gurdang Murrica owns them. Greatest country in the gurdang universe.

Normal Finetuning: Why is there no extraterrestrial life on Mars?

Rare word delexicalization + Multiple: Why is it illegal for a country to claim extraterrestrial land?

Input: Decriminalization means it's not legal, but you won't go to jail over it, like a speeding ticket. There's still a fine if you're caught doing it. Legalization means you're allowed to do it. I've come to know they are not the same; not sure how.

Normal Finetuning: What is the difference between decriminalization and legalization of marijuana?

Rare word delexicalization + Multiple: What is the difference between decriminalization and legalization?

Input: Because percentage growth is easier when you 're starting with a lower GDP . They 're leapfrogging decades of development and people are moving right from subsistence farming in small villages to a modern economy . They 're seeing the benefits of introducing electricity , telecommunications , modern agriculture , medicine , etc ... all at once . They still have large numbers of people with very low income levels that can catapult up the productivity ladder .Their GDP annual growth rates are clearly outperforming the US and UK economies . I am not very educated in economics , but am wondering how this could be so . I remember reading somewhere that it was state investment . Could this be true ? Thank you .

Normal Finetuning: How is China's economy growing so fast ?

Rare word delexicalization + Multiple: How is it possible that a country with such low GDP can have such high growth rates

---

Input: The global economy is n't the same as a couple of generations ago - it is globalised to a fuller extent . Attributes such as the renewed ease of international investment and trade have had a huge impact of how countries grow today .

Normal Finetuning: How did the global economy grow so much faster than it did in the 20th century

Rare word delexicalization + Multiple: How did the global economy grow so much faster than it did a couple of generations ago

---

Input: The court wants to avoid "playing to the public" by lawyers and wants to focus the questioning and presentations on purely legal arguments.

Normal Finetuning: Why are Supreme Court Justices sequestered?

Rare word delexicalization + Multiple: Why do we have to ask questions in court if we already know what's going to happen?

Input: It's basically unheard of for yakuza to kill regular people. Like rarely they will kill other yakuza in some power struggle thing but event hat is real real rare compared to the really killing based nature of a lot of US gangs. They aren't even strictly illegal organizations and basically operate openly. Some of them do illegal stuff for sure but they aren't even strictly criminal organizations and a ton of the stuff they do is just sketchy or unseemly rather than law breaking.

Normal Finetuning: Why are yakuza such a big deal in Japan?

Rare word delexicalization + Multiple: How is the yakuza not considered a criminal organization?

Input: It was the cold war, and making sure the food supply was managed properly in an attack situation was important.

Normal Finetuning: Why did the U.S. declare war on Japan in the first place?

Rare word delexicalization + Multiple: What was the importance of having a well-stocked army?

Input: It depends. If the artist self publishes they get about 0.60 cents. However if its under a major label they get as little as 0.08 cents per sale.

Normal Finetuning: How much does it cost an artist to release a song on iTunes?

Rare word delexicalization + Multiple: How do artists make money off of their music?

Table 17: Qualitative Examples.

ACL 2023 Responsible NLP Checklist

A For every submission:

☐ A1. Did you describe the limitations of your work?
Left blank.

☐ A2. Did you discuss any potential risks of your work?
Left blank.

☐ A3. Do the abstract and introduction summarize the paper's main claims?
Left blank.

☐ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☐ Did you use or create scientific artifacts?

Left blank.

☐ B1. Did you cite the creators of artifacts you used?
Left blank.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Left blank.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Left blank.

☐ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Left blank.

C ☐ Did you run computational experiments?

Left blank.

☐ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Left blank.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☐ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Left blank.

☐ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Left blank.

☐ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Left blank.

D ☐ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Left blank.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Left blank.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Left blank.