ECG-QALM: Entity-Controlled Synthetic Text Generation using Contextual Q&A for NER

Karan Aggarwal
Amazon
Seattle, WA
kagg@amazon.com

Henry Jin*
Harvard University
Cambridge, MA
helinjin@g.harvard.edu

Aitzaz Ahmad
Amazon
Seattle, WA
aitzaza@amazon.com

Abstract

Named Entity Recognition (NER) state-of-the-art methods requires high-quality labeled datasets. Issues such as scarcity of labeled data, under-representation of entities, and privacy concerns with using sensitive data for training, can be significant barriers. Generating synthetic data to train models is a promising solution to mitigate these problems. We propose ECG-QALM, a contextual question and answering approach using pre-trained language models to synthetically generate entity-controlled text. Generated text is then used to augment small labeled datasets for downstream NER tasks. We evaluate our method on two publicly available datasets. We find ECG-QALM is capable of producing full text samples with desired entities appearing in a controllable way, while retaining sentence coherence closest to the real world data. Evaluations on NER tasks show significant improvements (75% - 140%) in low-labeled data regimes.

1 Introduction

NLP tasks typically require large amounts of high-quality labeled data to train sufficiently accurate and useful models. However, in many domains, such as finance and healthcare, access to labeled data is often limited. In these domains, annotating data often requires strong domain expertise and therefore, crowdsourcing of labeled data is infeasible. The cost of annotating data by training an expert workforce is often too high for feasibility.

A small collection of labeled data also runs the risk of bias creeping in the data and may result in algorithms and models that reflect or even exploit this inherent bias. It also degrades the capability of models to generalize as small datasets are much less likely to have population groups or patterns under-represented (Zhou and Bansal, 2020). These issues need solutions that can perform well in low-labeled data regimes while combating data bias.

Synthetic data generation presents a promising solution to address the issues outlined above (Bayer et al., 2021). By synthetically generating data, we can augment small labeled datasets to build a training set. Synthetic data generation can also reduce bias in the data by to sufficiently represent all population groups. In particular, the field of controlled synthetic text generation has received increased attention in recent years. Controlled text generation provides the ability to control for traits such as tone, sentiment, and topic in the generation of a language model (Wang and Wan, 2018; Zeng et al., 2021). This lends controlled synthetic text generation as a useful technique for augmenting small or privacy-sensitive datasets. However, there has been limited work on the topic of entity-controlled synthetic text generation, *i.e.*, the task of generating coherent text while controlling for the named entities that appear in the generation (Dong et al., 2021).

In this paper, we study the problem of entity-controlled synthetic text generation. We propose, ECG-QALM, a ***E***ntity ***C***ontrolled Text ***G***eneration with Contextual ***Q***uestion ***A***nswering based pre-trained ***L***anguage ***M***odel, that can produce coherent text which contains specific entity tokens, generated in an order provided by the user. We are motivated by the need to synthetically augment datasets to improve performance on downstream NER tasks (Zhou et al., 2022). ECG-QALM provides multiple advantages. It is more sample efficient than other methods, as the model is trained on each block of each sample, unlike just seeing a sample in whole for Seq2Seq models like Dong et al. (2021); b) ECG-QALM sees a block of text which is relatively smaller than whole sample, prompted on entity to be inserted and conditioned on previous generation allowing for generation of more coherent text as demonstrated by generation metrics like perplexity versus SOTA Seq2Seq baselines; and c) unlike prior Seq2Seq methods like RNN (Dong et al., 2021) or using a vanilla GPT, where length of

*This work was done during Henry's internship at Amazon

Findings of the Association for Computational Linguistics: ACL 2023, pages 5649–5660
July 9-14, 2023

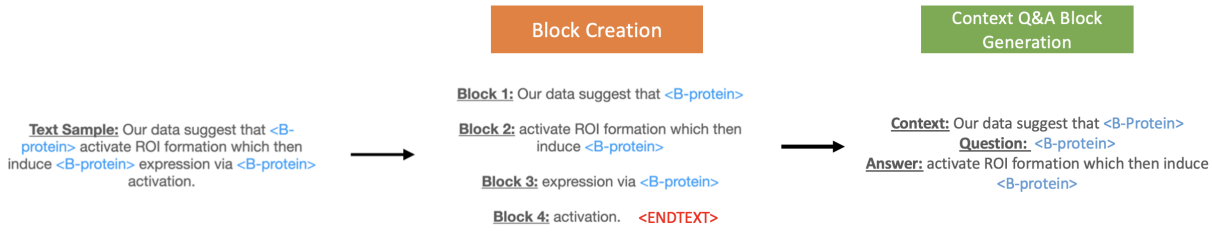

Figure 1: This figure depicts how a text sample with three <B-protein> entities is processed into blocks to train ECG-QALM. An <ENDTEXT> token always defines the final block of a decomposed text sample. In the Contextual Q&A Block Generation step, we show an example of a training sample for Block 3.

text generated is limited to 512/1024, ECG-QALM can generate as many blocks of (maximum) length 1024, as the number of entities to be inserted.

We make the following contributions: 1) we propose a novel approach using pre-trained language models to generate entity-controlled blocks of text, which can be chained to produce full synthetic text samples; 2) our method is capable of generating texts semantically closest to the training data while being distinct; and, 3) evaluations on publicly available datasets on NER task show a significant improvement in data augmentation performance for low-labeled data regimes, even by just using a purely synthetic data.

2 Related Work

Controlled text generation These methods control a certain aspect of generated text (Yang and Klein, 2021; Chan et al., 2020; Pascual et al., 2021) like sentiment (Wang and Wan, 2018) or concepts (Zeng et al., 2021). These methods focus a macro level aspect of the generated text while we want to control a fine grained text generation.

Data-to-text generation The idea is to convert a given set of words or structured data from tables into a piece of text. Most popular problem is table summary generation, also called table-to-text (Liu et al., 2018; Parikh et al., 2020; Chen et al., 2021) or keyword to text methods (Pascual et al., 2021; Tan et al., 2021). While similar, the key difference is they have a fixed set of entities in every generation.

Entity-controlled generation Works in the intent detection and slot filing literature for conversational systems have attempted entity-controlled generation (Jolly et al., 2020). Recently, Rosenbaum et al. (2022), attempted to use a pre-trained language model with an instruction prompt that uses examples as input in the prompt for model to generate synthetic text. Note, these models have been built in context of conversational systems and hence, have a goal to respond to a specific query which generating the output text, unlike our task of generating text with specified input entities.

Dong et al. (2021) proposed a solution to this exact problem for generating text with given entity types and their mentions, using a RNN based Seq2Seq architecture. Our method uses a pre-trained language model with a block-by-block generation mechanism, producing superior text over theirs. They do not evaluate on a downstream task like NER, unlike our work.

Data Augmentation for Named Entity Recognition These methods rely on substitution of entities in a given example with entity of same type to create examples. (Dai and Adel, 2020) proposed a simple random replacement which was further enhanced using language modeling to exploit context (Zhou et al., 2022; Ding et al., 2020). While these methods need seed text to generate each example, our method only needs entity tags to generate an example.

3 Methodology

We use a contextual question and answering based training approach to generate blocks of text with desired entity tags. This approach is able to reliably generate augmented text samples while retaining sentence coherence. Our method generates blocks of text delimited by entities to be inserted, and chaining these generated blocks to create full text samples. We use a GPT-2 language model in place of a recurrent network used by Dong et al. (2021)

Table 1: Dataset Statistics

Dataset	#Training	#Val	#Test	Avg. #Words	#Entities
JNLPBA	18606	1938	4259	22.97	5
BC5CDR	4561	4582	4798	24.93	2

Table 2: Generation Quality Metrics for the two datasets: Perplexity, Distinctness-3 (tri-gram), and Rouge-L. Cells with best score are highlighted in blue. (↑): higher the better; (↓): lower the better.

	JNPBLA (10%)							BC5CDR (10%)						
Metric	Gold Data	RS	EntInj	DACA	MELM	ECG-LM	ECG-QALM	Gold Data	RS	EntInj	DACA	MELM	ECG-LM	ECG-QALM
Perplexity(↓)	400.36	605.75	796.5	556.93	519.24	518.09	488.56	388.42	5856.61	1521.3	884.53	692.06	510.98	477.66
Distinctness-3(↑)	0.74	0.82	0.2	0.82	0.82	0.60	0.58	0.72	0.92	0.06	0.92	0.92	0.67	0.59
Rouge-L(↓)	1.0	0.72	0.30	0.72	0.72	0.39	0.20	1.0	0.83	0.26	0.83	0.83	0.25	0.21

to take advantage of using pre-trained large language models. The intuition being that using a pre-trained model helps in increasing diversity of the generated text.

3.1 Training

We first preprocess real world training text samples into blocks, whereby each block is composed of non-entity tokens and ends with an entity tag as shown in Figure 1. Every text sample is then decomposed into these blocks of text. An end of text token is added at the end. Therefore, a full text sample generation consists of chaining generated blocks until a block with an <ENDTEXT> token appears. Side benefit of creating blocks is increased number of (shorter, manageable) training examples that are easier to learn on, unlike existing methods that input entire text at once.

After decomposing text samples into such blocks, we arrange blocks into the question and answering format, which consists of three segments: context, question and answer. The context segment provides preceding text blocks, the question segment prompts the model for the desired token, and the answer block is the desired generation.

Context section consists of all blocks in the text sample, preceding the current block. This was motivated by the need for the model to be aware of the context for successive generation. The generation of each block must be a continuation of preceding blocks to maintain sentence level coherence.

Question segment prompts for the desired entity to appear in the next block. Therefore, through this prompting mechanism we control the desired entity tag to be generated. Following the "Question: " tag is a single token representing the desired entity.

Answer segment contains the desired text block to be generated. The final token in this block will therefore be the same token as in the question segment. With this three segment format, every block from the corpus represents a training sample for the language model.

3.2 Generation during Inference

At inference time, ECG-QALM generates text conditioned on two segments of context and question. To generate the first block, the context segment is blank, while the question segment contains the desired token to be generated in the first block. The model then completes the answer segment with a generated block, which is inserted into the context segment for the next block generation. A full text sample then is produced by concatenating blocks until an <ENDTEXT> token. If the desired entity tag does not appear in the generated block, we re-generate the block text until the tag appears.

3.3 Metrics

To evaluate the generated text, we quantitatively measure the quality of generation and performance on NER task. We use three generation quality metrics used in prior literature (Dong et al., 2021)[1]. **Perplexity** measures the 'surprisingness' of the generated text evaluated on a GPT model (Radford et al., 2018). **Distinctness** (Li et al., 2015) measures the uniqueness of tri-grams in the corpus. **Rouge-L** (Lin, 2004): One trivial sanity check is *regurgitation*, *i.e.,* if the generation model is simply memorizing the training set. Rouge-L score measures the similarity of the generated text with the training data by calculating the longest common sub-strings. Rouge-L score should be low, if the model is not just spitting out the training examples. A lower Rouge-L score indicates that the generated data is not trivially similar to the training data, hence, ensuring privacy complaint models by not regurgitating the private training data.

4 Experiments

We evaluate our model on two datasets described in Table 1. We compare with the following baselines:

Gold Data: Refers to the real world training data.

[1]Grammaticality (Warstadt et al., 2019) used in prior works is not a general metric as it is based on English literature.

Table 3: Macro F_1 scores on NER. Method with highest F_1 scores among the generation methods is **boldfaced** while method with highest F_1 score overall is indicated by blue. Δ is percentage difference in F_1 scores of gold data and ECG-QALM (w/ augmentation). (*) indicates statistically significant increase with student t-test (p<0.01).

			Generated Data						Gold Data Augmented with Generated Data						
Training Data	#Samples	Gold Data	RS	EntInj	DACA	MELM	ECG-LM	ECG-QALM	RS	EntInj	DACA	MELM	ECG-LM	ECG-QALM	Δ
JNLPBA (1%)	186	0.311	0.172	0.087	0.184	0.229	0.273	**0.347**	0.392	0.313	0.425	0.459	0.482	**0.546**	75.5%*
JNLPBA (10%)	1860	0.722	0.375	0.219	0.525	0.552	0.596	**0.641**	0.709	0.484	0.722	0.723	0.722	**0.723**	0.1%
BC5CDR (1%)	45	0.192	0.209	0.193	0.220	0.239	0.262	**0.283**	0.330	0.264	0.377	0.378	0.403	**0.463**	142.1%*
BC5CDR (10%)	456	0.749	0.587	0.424	0.689	0.711	0.734	**0.741**	0.711	0.738	0.744	0.758	0.757	**0.760**	1.4%*

RS (Dai and Adel, 2020): Simple data generation by substitution of named entities with entities of same type from the training gold samples.

DACA (Ding et al., 2020): Substitution method using a LSTM based language model to replace entities in the gold samples, by exploiting context.

MELM (Zhou et al., 2022): Substitution method using Masked Entity Language Model with XLM-Roberta with linearization, for a richer context.

EntInj (Dong et al., 2021): Text generation method based on LSTM Seq2Seq model. Closest work to ours, as it performs actual text generation.

ECG-LM: This is our own baseline Seq2Seq method, which generates the entire text given a list of entities, without a block-by-block generation.

Note: Generated text length in DACA, MELM, EntInj, and ECG-LM is limited by number of tokens model can generate (512/1024) at once; ECG-QALM is not, as it chains the generated blocks.

4.1 Experimental Settings

We use the training, validation, and testing data splits provided publicly in the datasets on Huggingface[2]. We use the training dataset (and its mentioned subsets) for training both the text generation models as well as training the downstream NER model. We use BERT (Devlin et al., 2018) for downstream NER task. NER results are reported on the complete test set for both the datasets.

We use an instance of OpenAI's GPT-2 (Radford et al., 2019) for ECG-QALM. Our model is trained with the Adam optimizer on a learning rate of 1e-3, one hundred warm-up steps, and an epsilon of 1e-8. The default CrossEntropy loss function is used, and the model is trained for up to 100 epochs. For the NER task, we train the BERT model for upto 10 epochs with a learning rate of 2e-3. These parameters were set based on hyper-parameter tuning on the validation set. *During generation, we exactly mimic the entity distribution of the gold data.* We can also change the entity distribution to boost under-represented entities as shown in Appendix A.1.

[2] https://huggingface.co/

5 Results and Discussion

5.1 Generation Quality

Generation quality results are shown in Table 2. We clearly observe that our method is lower on all three metrics against the original dataset, which is expected as ours is synthetically generated data. Our method works better than the only other text generation baseline EntInj (Dong et al., 2021) on all three metrics across the two datasets. Particularly, for the BC5CDR dataset, we note EntInj tends to generate repetitive text. *The correct benchmark are the substitution based baselines as our method inserts the entities in the same fashion*. We observe for the substitution based baselines, distinctness is highest, as expected as we have swapped commonly occurring trigram entities, while the perplexity is worse than ECG-QALM. This shows that swapping affects the lexical meaning of the text, even when done intelligently in DACA/MELM. While we also insert randomly chosen entities in our generated text, these results indicate that our method generates coherent generic text where semantic meaning of the type of the entity is preserved.

Our generated data has the lowest Rouge-L scores. Hence, our generated data is not simply memorizing the training data, it is quite different than the gold data. We can see the huge gap with the substitution methods; while the data from substitution methods is practically same as the gold data, ours is distinct. Based on these metrics, *we can claim that generated text is semantically closest to the original corpus, while being distinct.*

5.2 Named Entity Recognition Task

We took two subsets of the JNLPBA and BC5CDR datasets: 1% and 10% as we found the performance on datasets was already saturated at their full sizes

as number of samples was enough. Hence, we present the results on first 1% and 10% examples of training splits to show the comparisons. We present two settings: (a) w/o augmentation with gold data; and (b) augmentation with gold data. Generated text for all methods is same size as gold data. Note, no changes were made to test/val sets.

Table 3 shows the results for the two subsets of the two datasets. From the results five things stand out: 1) Augmenting gold data with our synthetically generated data always out-performs a model trained with the gold data; 2) using **only** synthetically generated data is comparable in performance to the gold data in medium labeled setting (10%) ; 3) our synthetically generated data outperforms gold data in low labeled data setting (1%) subsets; 4) our synthetically generated data gives better performance vs all baseline methods; and 5) our novel block-by-block generation approach significantly improves over a vanilla GPT-2 (ECG-LM) model.

Our finding that synthetically generated data can get us a comparable performance to gold data has an application in making the models trained for downstream tasks like NER, privacy preserving, as they do not have to be trained on the real data. This finding can be attributed to zero/few-shot capabilities of large language models (Wei et al., 2021). Hence, the capability to produce texts that can generalize better on unseen test set while other models are only able to capture subset of test set distribution reflected in the training gold dataset. Our results show our method of generation can be *quite effective as a data augmentation method in a low labeled data regime*.

5.3 Generating more text in low resource

Previously, we only showed the results by generating synthetic data of same size as the gold data. We perform an experiment to see if there is further improvement in the performance as we add more generated data with the JNLPBA (1%) dataset. We observe that F_1 score keeps improving going up to 0.70 vs gold data at 0.31 in Figure 2. Note, we only use the entity mentions found in the JNLPBA (1%) dataset to fill in the entity tags in the generated text. This is remarkable considering that 10x real data for JNLPBA (10%) has a F_1 score of 0.72. This is a further evidence that our model is able to generate text that is similar to real data.

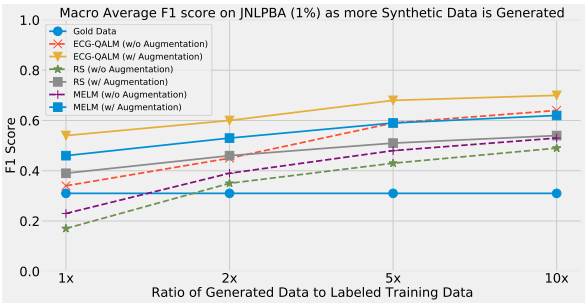

Figure 2: Macro Average F1 score as we augment more generated data to the JNLPBA (1%) dataset.

6 Conclusion

Synthetic data generation is a promising approach to train large language models in order to deal with scarcity of labeled data. In this work, we study the problem of conditional text generation where the conditions are provided as a list of entities that must appear in the text in a manner desired by the user. We propose ECG-QALM that can generate blocks of text conditioned on the desired entities. We test our generation system on generation quality metrics and NER task. Evaluations show that our method outperforms baselines in terms of both generation quality and NER performance. Our block-by-block generation provides significant gains over using a fine-tuned vanilla LLM for generation.

7 Limitations

The major limitations of this work are:

- We show results on two public datasets, from bio-medical and bio-chemical domains. These results may not generalize to other domains.
- Our results indicate benefit in low resource settings, while no appreciable benefit is seen for medium or high resource settings.
- Our method relies on GPT-2, a large language model that needs humongous compute resources and a long training time. It takes about 2 hours to generate 50 samples, versus the baselines like vanilla GPT-2 (ECG-LM) taking 30 mins or EntInj taking about 10 mins to generate same number of examples with much less memory requirements.
- We use quantitative measures to evaluate the quality of text generation, which might not be enough to capture the quality of generated text. Gold standard of measuring the quality is human evaluation, which is expensive.

References

Markus Bayer, Marc-André Kaufhold, and Christian Reuter. 2021. A survey on data augmentation for text classification. *ACM Computing Surveys*.

Alvin Chan, Yew-Soon Ong, Bill Pung, Aston Zhang, and Jie Fu. 2020. Cocon: A self-supervised approach for controlled text generation. *arXiv preprint arXiv:2006.03535*.

Wenqing Chen, Jidong Tian, Yitian Li, Hao He, and Yaohui Jin. 2021. De-confounded variational encoder-decoder for logical table-to-text 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 1: Long Papers)*, pages 5532–5542.

Xiang Dai and Heike Adel. 2020. An analysis of simple data augmentation for named entity recognition. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3861–3867.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Bosheng Ding, Linlin Liu, Lidong Bing, Canasai Kruengkrai, Thien Hai Nguyen, Shafiq Joty, Luo Si, and Chunyan Miao. 2020. DAGA: Data augmentation with a generation approach for low-resource tagging tasks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6045–6057, Online. Association for Computational Linguistics.

Xiangyu Dong, Wenhao Yu, Chenguang Zhu, and Meng Jiang. 2021. Injecting entity types into entity-guided text generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 734–741, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shailza Jolly, Tobias Falke, Caglar Tirkaz, and Daniil Sorokin. 2020. Data-efficient paraphrase generation to bootstrap intent classification and slot labeling for new features in task-oriented dialog systems. In *Proceedings of the 28th International Conference on Computational Linguistics: Industry Track*, pages 10–20.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Tianyu Liu, Kexiang Wang, Lei Sha, Baobao Chang, and Zhifang Sui. 2018. Table-to-text generation by structure-aware seq2seq learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Ankur P Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. 2020. Totto: A controlled table-to-text generation dataset. *arXiv preprint arXiv:2004.14373*.

Damian Pascual, Beni Egressy, Clara Meister, Ryan Cotterell, and Roger Wattenhofer. 2021. A plug-and-play method for controlled text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3973–3997, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

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

Andy Rosenbaum, Saleh Soltan, Wael Hamza, Yannick Versley, and Markus Boese. 2022. Linguist: Language model instruction tuning to generate annotated utterances for intent classification and slot tagging. *arXiv preprint arXiv:2209.09900*.

Bowen Tan, Zichao Yang, Maruan Al-Shedivat, Eric Xing, and Zhiting Hu. 2021. Progressive generation of long text with pretrained language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4313–4324, Online. Association for Computational Linguistics.

Ke Wang and Xiaojun Wan. 2018. Sentigan: Generating sentimental texts via mixture adversarial networks. In *IJCAI*, pages 4446–4452.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Kevin Yang and Dan Klein. 2021. Fudge: Controlled text generation with future discriminators. *arXiv preprint arXiv:2104.05218*.

Qingkai Zeng, Jinfeng Lin, Wenhao Yu, Jane Cleland-Huang, and Meng Jiang. 2021. Enhancing taxonomy completion with concept generation via fusing relational representations. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2104–2113.

Ran Zhou, Xin Li, Ruidan He, Lidong Bing, Erik Cambria, Luo Si, and Chunyan Miao. 2022. Melm: Data augmentation with masked entity language modeling for low-resource ner. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2251–2262.

Xiang Zhou and Mohit Bansal. 2020. Towards robustifying nli models against lexical dataset biases. *arXiv preprint arXiv:2005.04732*.

A Appendix

A.1 Ablation: Generating Under-represented Entities

We perform a simple experiment to see how ECG-QALM can potentially also be beneficial to generate data that could be augmented to boost the performance of under-represented entities in the original training data. To refresh, we kept the entity distribution exactly same as training data while generating data through our method. To boost the relative frequency of the under-represented entities, we generate examples proportional to the inverse frequency of the entities present.

Let the training data have n samples. Each sample has a set of named entities in it, *e.g.*, a sample containing the set of entities, {<B-Protein>, <B-DNA>, <B-DNA>}, has two distinct entities in it. We calculate the frequency of each named entities over the entire training corpus. Next, we calculate the score of each sample by adding the inverse frequency of each named entity in that sample. For example, if the <B-Protein> has a inverse frequency of 10 and <B-DNA> has a inverse frequency of 100, this sample would get a score of 210. Next, we normalize these scores by the sum of scores of every sample in the corpus. This gives us a probability score for using the entity set of a sample to be picked while generating. Entity set of a sample with a probability score of 1% would be picked 10 times while generating 1000 synthetic examples, for instance.

Hence, this ensures that under-represented entities are boosted in the new generated data. This could be used for augmenting the original data to improve performance on under-represented entities. Note, we can also generate random entity sets just with under-represented examples. However, we prefer not to do it as it could alter the co-occurrence of entities in the generated text, shifting the training set distribution so significantly that it no longer represents the original training set.

We take JNLPBA (10%) dataset for this measure, as it has a large number of entities. Results after generating a synthetic data of same size as the original training set are shown in Table 4. While there is a 1% increase in the macro average, we observe the performance over different entities are mixed. While there is generally an increase in the performance for the under-represented entities, there is a drop for selected entities like <B-RNA>, despite almost doubling of number of samples for the entity. For the abundant entities like <B-Protein> performance is similar. In future, it would be worthwhile to experiment with different distributions of co-occurrence of entities instead of deriving it from the gold (training) data.

A.2 Examples of Generated Text

In the section below we shows few examples of generated text by ECG-QALM and EntInj (Dong et al., 2021), the only text generation method in baselines. Our method generates semantically meaningful examples, while EntInj generates quite repetitive examples. Text highlighted in Red marks the entities.

A.2.1 ECG-QALM

The examples below seem grammatically correct, as was the observation over the entire generated corpus. However, as we randomly insert entity mentions after we generated the entity tags, most of the generated examples are not factual. E.g., DTG is not associated with treatment of blood clotting as generated in the first example. Our goal was not factual correctness but ensuring that the generated data preserves the distribution of the training data, which seems to be the case based on generation metrics and results on NER task.

> The efficacy of DTG in the treatment of impaired blood clotting likewise did not appear to be affected by the rate of administration, although no formal statistical comparisons were made .

> The prevalence rate for death was the most important reason for preference, cited by 67 . 3 % of patients preferring Picloxydine and 54 . 2 % of patients who preferred a p < or = 0 . 001) .

> The reduction of acetaminophen at 1 and 4 days after gestation not glomeruli with ataxic movements than control rats .

Table 4: Macro F_1 scores on the NER task for gold data, our generated data, our generated data with Under-represented generation (+URG) for JNLPBA (10%) dataset. Cells with highest (almost highest) F_1 score for an entity (row) are highlighted in blue. Second highest value is underlined.

Training Data (→) / Entity(↓)	% original frequency	% generated frequency	W/o Augmentation: Gold Data	W/o Augmentation: ECG-QALM	W/o Augmentation: ECG-QALM(+URG)	W/ Augmentation: ECG-QALM	W/ Augmentation: ECG-QALM(+URG)
JNLPBA (10%) [#training samples=1860]							
B-DNA	8.53%	8.45%	0.70	0.65	0.65	0.71	0.71
B-RNA	0.85%	1.92%	0.69	0.64	0.68	0.70	0.68
B-cell-line	3.45%	5.57%	0.51	0.46	0.45	0.51	0.53
B-cell-type	6.23%	6.69%	0.69	0.59	0.60	0.70	0.70
B-protein	27.98%	20.38%	0.76	0.76	0.64	0.75	0.76
I-DNA	14.15%	14.04%	0.78	0.68	0.69	0.79	0.79
I-RNA	1.35%	2.43%	0.82	0.70	0.81	0.81	0.83
I-cell-line	6.66%	8.92%	0.55	0.46	0.45	0.53	0.54
I-cell-type	8.18%	8.20%	0.69	0.63	0.63	0.71	0.70
I-protein	22.58%	20.15%	0.77	0.69	0.69	0.77	0.77
Macro Avg.			0.72	0.64	0.66	0.72	0.73

The aims of this study were to confirm our previous findings in a separate cohort of patients and to determine the time course of the cardiovascular consequences of stopping sertraline in the expectation that this might shed light on the mechanisms by which the mechanisms by Tamoxifen is being a significant reduction of the activity on the drug causes the sodium associated with cephalothin sodium associated with povidone - iodine is associated with cocaine and inhibition with the use of tuberculosis and area in this effect.

MR imaging with quantitative diffusion mapping of E4031 (0 . g), p - choloroaniline) and outcome in organ transplant controls, and / L and the development of blood coagulation by a potential is also more than the development of systolic dysfunction and possibly .

The possibilities that these findings might be the result of non - induced <> is a result of monoamine oxidase or inhibition of monoamine

A.2.2 EntInj (Dong et al., 2021)

We observed a lot of repetition in the generated text by EntInj method. This looping behavior is shown in Example 2 and 3 below. Note, unlike our method, EntInj has access to the exact same entity mentions as they appear in the training data, having an inherent advantage with this additional information.

<unk> : The cardiovascular responses to standing and standing . 4 patients were studied in the drug . 4 days . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 . 4 .

telithromycin - induced bromo tetrahydropyranyladriamycin pituitary carsinom agitation one : a longitudinal study of

In the study was undertaken to the combination of painful , headache , bleed , which was only induced by epilepticus drug , and bronchitis

Investigation of anti - inflammatory agents are warranted in the caudate nucleus . injection of Allopurinol injection of bacterial collagenase - induced

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Sec 7

☒ A2. Did you discuss any potential risks of your work?
Our method does not pose a risk that we can think of.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and Sec 1

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

B ☑ Did you use or create scientific artifacts?

Sec 5

☑ B1. Did you cite the creators of artifacts you used?
Sec 4.1

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. 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)?
Not applicable. 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?
Not applicable. Left blank.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Sec 4

☑ 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.
Sec 4

C ☑ Did you run computational experiments?

Sec 4.1

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
We used opensource GPT-2 model

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?
Sec 4.1

☑ 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?
Sec 5

☑ 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.)?
Sec 4.1

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.?
No response.

☐ 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)?
No response.

☐ 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?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.