Mask-then-Fill: A Flexible and Effective Data Augmentation Framework for Event Extraction

Jun Gao[1,3*] **Changlong Yu**[4] **Wei Wang**[5] **Huan Zhao**[6] **Ruifeng Xu**[1,2,3†]

[1]Harbin Institute of Technology (Shenzhen) [2]Peng Cheng Laboratory

[3]Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies

imgaojun@gmail.com xuruifeng@hit.edu.cn

[4]HKUST, Hong Kong, China [5]Tsinghua University [6]4Paradigm. Inc.

cyuaq@cse.ust.hk weiwangorg@163.com zhaohuan@4paradigm.com

Abstract

We present **Mask-then-Fill**, a flexible and effective data augmentation framework for event extraction. Our approach allows for more flexible manipulation of text and thus can generate more diverse data while keeping the original event structure unchanged as much as possible. Specifically, it first randomly masks out an adjunct sentence fragment and then infills a variable-length text span with a fine-tuned infilling model. The main advantage lies in that it can replace a fragment of arbitrary length in the text with another fragment of variable length, compared to the existing methods which can only replace a single word or a fixed-length fragment. On trigger and argument extraction tasks, the proposed framework is more effective than baseline methods and it demonstrates particularly strong results in the low-resource setting. Our further analysis shows that it achieves a good balance between diversity and distributional similarity.

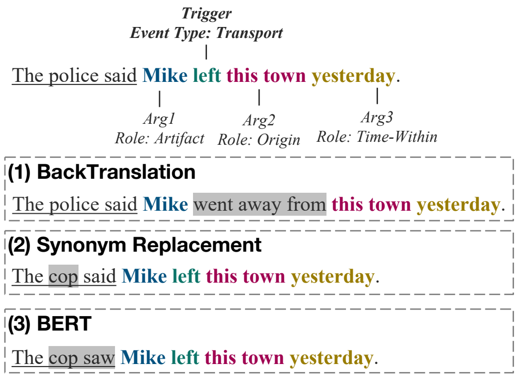

Figure 1: Visualization of three different data augmentation methods applied to a sentence containing a "Transport" event. Spans marked with different colors are event triggers and arguments. The parts of the augmented sample that differ from the original are colored in gray. **Backtranslation** (Xie et al., 2020) translates the input sentence into another language and back to the original. **Synonym Replacement** (Dai and Adel, 2020) and **BERT** (Yang et al., 2019) replace words in the sentence.

1 Introduction

Event Extraction (EE), which aims to extract triggers with specific types and their arguments from unstructured texts, is an important yet challenging task in natural language processing. In recent years, deep learning methods have emerged as one of the most prominent approaches for this task (Nguyen and Nguyen, 2019; Lin et al., 2020; Du and Cardie, 2020; Paolini et al., 2021; Lu et al., 2021; Lou et al., 2022). However, they are notorious for requiring large labelled data, which limits the scalability of EE models. Annotating data for EE is usually costly and time-consuming, as it requires expert knowledge. One possible solution is to leverage data augmentation (DA) (Simard et al., 1998).

Existing DA methods for NLP can be broadly classified into two types: (1) the first is to augment training data by modifying existing examples (Sennrich et al., 2016; Şahin and Steedman, 2018; Dai and Adel, 2020; Wei and Zou, 2019), and (2) the second is to generate new data by estimating a generative process and sample from it (Anaby-Tavor et al., 2020; Quteineh et al., 2020; Yang et al., 2020; Ye et al., 2022). Since the EE task requires DA methods to generate augmented samples and token-level labels jointly, the second type of DA method is inapplicable here. In this study, we mainly focus on the first type of method.

Applying existing DA methods to the EE task is more challenging than to translation or classification tasks, because we need to augment training data while keeping the event structure (trigger and arguments) unchanged. Figure 1 presents examples of three different DA methods applied to a sentence containing a "*Transport*" event. The event is triggered by word "*left*" and it has three arguments with different roles ("*Mike*", "*this town*"

* Work done when Jun Gao was interning at 4Paradigm

† Corresponding author

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 4537–4544

and "*yesterday*"). As shown in Figure 1, it is infeasible to apply sentence-level DA methods such as BackTranslation (Xie et al., 2020), because it may change the event structure (change "*left*" to "*went away from*"). Previous attempts on DA for such tasks typically use heuristic rules such as synonym replacement (Dai and Adel, 2020; Cai et al., 2020) or context-based words substitution with BERT (Yang et al., 2019). Their idea is to replace adjunct tokens (the tokens in sentences except triggers and arguments) with other tokens, and thus can ensure the event structure is unchanged as much as possible. However, recent studies (Ding et al., 2020; Yang et al., 2020; Ye et al., 2022) find that such methods provide limited data diversity. In NLP, the diversity of data is mainly reflected in two aspects: expression diversity and semantic diversity (Zhao et al., 2019). The Synonym Replacement and BackTranslation methods lack semantic diversity, because they can only produce samples with similar semantics. The BERT-based method can only replace words and cannot change the syntax, so it cannot generate samples with a wide variety of expressions. The lack of sufficient diversity may lead to greater overfitting or poor performance through training on examples that are not representative.

To this end, we present **Mask-then-Fill**, a flexible and effective data augmentation framework for event extraction. Our approach allows for more flexible manipulation of text and thus can generate more diverse data while keeping the original event structure unchanged as much as possible. Specifically, we first define two types of text fragments in a sentence: event-related fragments (trigger and arguments) and adjunct fragments (e.g. "*The police said*"). Then, we model DA for the EE task as a **Mask-then-Fill** process: we first randomly masks out an adjunct sentence fragment and then infills a variable-length text span with a fine-tuned infilling model (T5) (Raffel et al., 2020). The main advantage lies in that it can replace a fragment of arbitrary length in the text with another fragment of variable length, compared to the existing methods which can only replace a single word or a fixed-length fragment.

To the best of our knowledge, we are the first to augment training data for event extraction via text infilling. We empirically show that the **Mask-then-Fill** framework improves performance for both classification-based (EEQA) and generation-based (Text2Event) event extraction models on a well-known EE benchmark dataset (ACE2005). Especially, it demonstrates strong results in the low resource setting. We further investigate reasons for its effectiveness by introducing two metrics, *Affinity* and *Diversity*, and find that the data augmented by our approach have better diversity with less distribution shifts, achieving a good balance between diversity and distributional similarity.

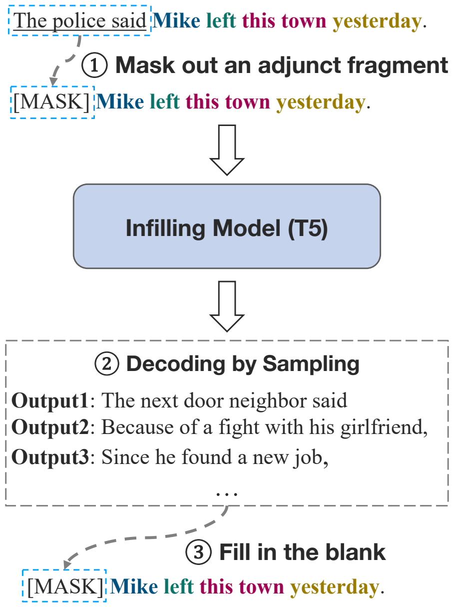

Figure 2: Overview of the proposed Mask-then-Fill framework.

2 The Mask-then-Fill Framework

Figure 2 presents an overview of Mask-then-Fill framework. The input sentence contains two types of text fragments: event-related fragments (words with colors) and adjunct fragments (underlined). Our idea is to rewrite the whole adjunct fragment instead of replacing some words, and the rewritten sentence fragment should fit the context and should not introduce new events. To this end, we model DA for EE as a **Mask-then-Fill** process: we first randomly mask out an adjunct sentence fragment and then infills a variable-length text span with a fine-tuned infilling model. In the following, we describe in detail the Mask-then-Fill framework.

Mask out an adjunct fragment. Given a prototype sentence $\boldsymbol{X} = \{x_1, \cdots, x_L\}$ of length L from the training set, we first define an adjunct fragment as a set of non-overlapping spans of x that do not contain the event triggers and arguments. We then replace one of the adjunct fragments with a [MASK] symbol. The incomplete sentence $\hat{\boldsymbol{x}}$

is a version of $\boldsymbol{x}$ with a fragment replaced with a [MASK] symbol.

Blank Infilling Model. We formulate our blank infilling process as the task of predicting the missing span of text which is consistent with the preceding and subsequent text. Figure 2 gives an example with an incomplete input sentence $\widetilde{\boldsymbol{x}}$, where the [MASK] is a placeholder for a blank, which has masked out multiple tokens. Our goal is to predict only the missing span $\boldsymbol{y}$ which will replace the [MASK] token in $\widetilde{\boldsymbol{x}}$. Therefore, the infilling task can be cast as learning $p(\boldsymbol{y}|\widetilde{\boldsymbol{x}})$.

To train our infilling model, we fine-tune a pretrained sequence-to-sequence model T5 (Raffel et al., 2020) on the Gigaword corpus (Graff et al., 2003), which is from similar domains as the event extraction dataset ACE2005 adopted by our work. Given a corpus consisting of plain sentences, we first produce large pools of infilling examples and then train the T5 model on these examples. For a given complete sentence $\boldsymbol{x}$ from the training corpus, we generate an infilling example $\widetilde{\boldsymbol{x}}$ with the following procedure: (1) randomly sample a span length l from the range of $[1, \min(10, l)]$; (2) split the sentence into l spans; (3) randomly select a span to be replaced with a [MASK] symbol. The replaced span is used as the target $\boldsymbol{y}$. We then fine-tune the T5 model on these infilling examples, yielding the model of the form $p_\theta(\boldsymbol{y}|\widetilde{\boldsymbol{x}})$.

Fill in the blank. Once trained, the infilling model can be used to take the incomplete sentence $\widetilde{\boldsymbol{x}}$, containing one missing span, and return a predicted span $\boldsymbol{y}$. We then replace the [MASK] token in $\widetilde{\boldsymbol{x}}$ with the predicted span $\boldsymbol{y}$ to generate an augmented example. Note that we can produce large pools of augmented samples using top-k sampling.

3 Experimental Setup

Dataset. We empirically evaluate our proposed data augmentation method for event extraction on the ACE2005 corpus[1] with the same train-dev-test split and preprocessing step as previous works (Zhang et al., 2019; Wadden et al., 2019).

We simulate a low-resource setting by randomly sampling 1,000, 4,000 and 8,000 examples from the training set to create the small, medium, and large training sets (denoted as **S**, **M**, **L** in Table 1, whereas the complete training set is denoted as **F**). We only augment the training data and keep the dev set and test sets unchanged.

Evaluation Metrics. Following the previous works (Du and Cardie, 2020; Lu et al., 2021) on event extraction, we adopt the same evaluation criteria defined in Li et al. (2013): (i) An event trigger is correctly identified and classified (**Trig-ID+C**) if its offsets match a gold trigger and its event type is also correct. (ii) An argument is correctly identified and classified (**Arg-ID+C**) if its offsets and event type match a gold argument and its event role is also correct.

Event Extraction Models. In our study, we consider two representative models for event extraction:

- **Text2Event** (Lu et al., 2021) is a framework to solve the event extraction task by casting it as a SEQ2SEQ generation task. All triggers, arguments, and their labels are generated as natural language words.
- **EEQA** (Du and Cardie, 2020) formulates the event extraction task as a question answering task. They develop two BERT-based QA models – one for event trigger detection and the other for argument extraction.

Comparison Methods. We compare our proposed data augmentation method **Ours (t5-small)** with three baselines: (1) **Synonym Replacement** replaces adjunct tokens with one of their synonyms retrieved from WordNet (Miller, 1992) at random; (2) **BERT** replaces adjunct tokens with others randomly drawn according to the pretrained BERT's distribution; (3) **Span-BackTranslation**: Inspired by Yaseen and Langer (2021), we only "back translate" randomly selected adjunct spans to prevent the model from changing the event structure.

Hyperparameters. For all data augmentation methods, we tune the number of augmentation samples per training sample from a list of numbers: $\{1, 3, 6, 10\}$.

4 Results and Analysis

Main Results. The main results are presented in Table 1, where we use two EE models (**Text2EVent** and **EEQA**) to test the performance of different DA methods in both low-resource (**S**, **M** and **L**) and normal (**F**) settings. As shown in the table, we observe that Ours (t5-small) achieves the best overall

[1]https://catalog.ldc.upenn.edu/LDC2006T06

EE Model	DA Method	Trig-ID+C (F1)				Arg-ID+C (F1)			
		S	M	L	F	S	M	L	F
Text2Event	No Augmentation	45.44	59.75	63.55	67.06	22.05	36.04	40.35	49.30
	Synonym Replacement	49.14	61.96	63.73	**69.09**	27.71	39.64	43.63	48.95
	BERT	48.66	60.75	63.81	68.33	26.71	38.75	41.44	48.41
	Span-BackTranslation	47.91	61.54	64.59	67.58	26.68	38.18	43.39	47.93
	Ours (t5-small)	**52.32**	**63.38**	**67.25**	69.03	**28.68**	**39.79**	**44.73**	**50.29**
EEQA	No Augmentation	48.05	64.20	64.06	67.13	39.81	56.30	59.27	61.93
	Synonym Replacement	**54.86**	64.03	65.71	68.05	42.99	56.62	56.40	61.50
	BERT	53.61	63.23	65.90	68.35	38.80	52.82	**59.49**	**61.62**
	Span-BackTranslation	53.26	62.64	65.46	68.40	42.47	56.64	55.87	61.55
	Ours (t5-small)	54.80	**64.37**	**67.33**	**69.62**	**47.67**	**56.70**	58.52	**61.62**

Table 1: Results on trigger extraction and argument extraction using different subsets of the training data. The best results are marked in bold, and the second best is underlined.

performance among all DA methods on both trigger extraction (F1) and argument extraction (F1). Using our DA method gives the best results for the Text2event model on 7 out of 8 datasets. For the EEQA model, our method achieves the best results on 6 out of 8 datasets, where the difference between our method and the best results on **Trig-S** and **Arg-L** is very small, with only 0.06 and 0.97 points difference between them, respectively. Particularly, our methods demonstrates strong results in the low-resource setting. Using our DA gives the Text2Event model a performance improvement of 15.14% and 30.07% on **Trig-S** and **Arg-S**, respectively.

We also notice that as the amount of data increases, the improvement from all DA method decreases, and in some cases (EEQA model on **Arg-L** and **Arg-F**), there is even a slight decrease in performance. In the case of more data, the model may overfit if the augmented data are just some similar samples rather than data with large variations.

DA Method	Affinity	Dist-1	Dist-2
Synonym Replacement	-0.118	0.400	0.523
BERT	**-0.082**	0.374	0.496
Span-BackTranslation	-0.155	0.407	0.513
Ours (t5-small)	-0.086	**0.488**	**0.612**

Table 2: Results on *Affinity* and *Diversity*. The best results are marked in bold. The second best is underlined.

Affinity and Diversity. Inspired by Gontijo-Lopes et al. (2020), we further investigate reasons for its effectiveness by introducing two metrics, *Affinity* and *Diversity*, where *Affinity* quantifies how augmentation shifts data distribution and *Diversity* measures the complexity of the augmented data. We measure *Affinity* by computing the difference between the loss of a model trained on the original training set and tested on the original example, and the loss of the same model tested on an augmented example. We use the Dist-1/2 metric (Celikyilmaz et al., 2020), commonly used in text generation, to assess the *Diversity* of the augmented data. For implementation details of two metrics, see Appendix.

We first construct a new test set by generating a new sample for each data in the test set. We then calculate the *Affinity* and Dist-1/2 scores between the new data set and the original data set, respectively. As shown in Table 2, it is clear that the data augmented by our DA method have better diversity with less distribution shifts, obtaining a balance between diversity and distributional similarity.

Figure 3: Augmented examples of four different DA methods. Given a sentence containing an “Attack” event triggered by the word "war", we generate two new samples for each DA method. The parts of the new sample that differ from the original are colored in gray.

Case Study. Figure 3 presents examples generated by different DA methods. Given a sentence containing an “Attack” event triggered by the word "war", we generated two new samples for each DA method, and the parts of the new sample that differ from the original are colored in gray. Obviously, The synonym replacement based on WordNet cannot avoid introducing some words that do not fit the context (e.g “unify” and “DoS”), while the BERT-based word replacement can consider the context better. However, they both provide limited diversity. BackTranslation method performs even worse in terms of data diversity. Its generated data differs very little from the original sentence. Finally, compared with the original sentences, the new samples generated by our method are more fluent and more different in expression and semantics. Therefore, it not only generates data that fits the context better, but also provides better diversity.

5 Conclusion

In this paper, we present **Mask-then-Fill**, a flexible and effective data augmentation framework for event extraction. Our approach allows for more flexible manipulation of text and thus can generate more diverse data while keeping the original event structure unchanged. The main advantage lies in that it can replace a fragment of arbitrary length in the text with another fragment of variable length. We empirically show that the **Mask-then-Fill** framework improves performance for both **EEQA** and **Text2Event** EE models on the ACE2005 dataset. It demonstrates particularly strong results in the low-resource setting. Our further analysis shows that it achieves a good balance between diversity and distributional similarity.

Limitations

This paper presents a flexible and effective data augmentation framework for event extraction tasks. Here, we note some of **Mask-then-Fill** framework's limitations. First, performance gains can be marginal when data is sufficient. We believe this approach has much room for improvement in generating more diverse data. In this work, we select only one adjunct fragment at a time for modification, and modifying multiple adjunct fragments in an event mention can further enhance the diversity of the generated data. Second, currently this method can only replace one fragment at a time. This makes it easier to control the properties of the generated fragments, such as length or style. It is possible to modify multiple fragments at the same time using some existing techniques (Donahue et al., 2020; Du et al., 2022; Chen et al., 2022). This approach is more efficient, but it is prone to generate incoherent augmented samples and thus introduce more noise. A possible approach to solve this problem is to design some sample selection strategies.

Acknowledgements

This work was partially supported by the National Natural Science Foundation of China 62006062 and 62176076, Shenzhen Foundational Research Funding JCYJ20200109113441941, JCYJ20210324115614039, The Major Key Project of PCL2021A06, Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies 2022B1212010005.

References

Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. 2020. Do not have enough data? deep learning to the rescue! In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7383–7390. AAAI Press.

Udit Arora, William Huang, and He He. 2021. Types of out-of-distribution texts and how to detect them. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10687–10701, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hengyi Cai, Hongshen Chen, Yonghao Song, Cheng Zhang, Xiaofang Zhao, and Dawei Yin. 2020. Data manipulation: Towards effective instance learning for neural dialogue generation via learning to augment and reweight. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6334–6343, Online. Association for Computational Linguistics.

Asli Celikyilmaz, Elizabeth Clark, and Jianfeng Gao. 2020. Evaluation of text generation: A survey. *ArXiv preprint*, abs/2006.14799.

Yi Chen, Haiyun Jiang, Lemao Liu, Rui Wang, Shuming Shi, and Ruifeng Xu. 2022. Mcpg: A flexible multi-level controllable framework for unsupervised paraphrase generation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*.

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, Barcelona, Spain (Online). International Committee on Computational Linguistics.

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.

Chris Donahue, Mina Lee, and Percy Liang. 2020. Enabling language models to fill in the blanks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2492–2501, Online. Association for Computational Linguistics.

Xinya Du and Claire Cardie. 2020. Event extraction by answering (almost) natural questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 671–683, Online. Association for Computational Linguistics.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. Glm: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335.

Raphael Gontijo-Lopes, Sylvia J Smullin, Ekin D Cubuk, and Ethan Dyer. 2020. Affinity and diversity: Quantifying mechanisms of data augmentation. *ArXiv preprint*, abs/2002.08973.

David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2003. English gigaword. *Linguistic Data Consortium, Philadelphia*, 4(1):34.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 73–82, Sofia, Bulgaria. Association for Computational Linguistics.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7999–8009, Online. Association for Computational Linguistics.

Chenwei Lou, Jun Gao, Changlong Yu, Wei Wang, Huan Zhao, Weiwei Tu, and Ruifeng Xu. 2022. Translation-based implicit annotation projection for zero-shot cross-lingual event argument extraction. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2076–2081.

Yaojie Lu, Hongyu Lin, Jin Xu, Xianpei Han, Jialong Tang, Annan Li, Le Sun, Meng Liao, and Shaoyi Chen. 2021. Text2Event: Controllable sequence-to-structure generation for end-to-end event extraction. 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 2795–2806, Online. Association for Computational Linguistics.

George A. Miller. 1992. WordNet: A lexical database for English. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*.

Trung Minh Nguyen and Thien Huu Nguyen. 2019. One for all: Neural joint modeling of entities and events. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6851–6858. AAAI Press.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cícero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Husam Quteineh, Spyridon Samothrakis, and Richard Sutcliffe. 2020. Textual data augmentation for efficient active learning on tiny datasets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7400–7410, Online. Association for Computational Linguistics.

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

Gözde Gül Şahin and Mark Steedman. 2018. Data augmentation via dependency tree morphing for low-resource languages. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5004–5009, Brussels, Belgium. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Patrice Y Simard, Yann A LeCun, John S Denker, and Bernard Victorri. 1998. Transformation invariance in pattern recognition—tangent distance and tangent propagation. In *Neural networks: tricks of the trade*, pages 239–274. Springer.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. 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 5784–5789, Hong Kong, China. Association for Computational Linguistics.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. 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 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Qizhe Xie, Zihang Dai, Eduard H. Hovy, Thang Luong, and Quoc Le. 2020. Unsupervised data augmentation for consistency training. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Sen Yang, Dawei Feng, Linbo Qiao, Zhigang Kan, and Dongsheng Li. 2019. Exploring pre-trained language models for event extraction and generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5284–5294, Florence, Italy. Association for Computational Linguistics.

Yiben Yang, Chaitanya Malaviya, Jared Fernandez, Swabha Swayamdipta, Ronan Le Bras, Ji-Ping Wang, Chandra Bhagavatula, Yejin Choi, and Doug Downey. 2020. Generative data augmentation for commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1008–1025, Online. Association for Computational Linguistics.

Usama Yaseen and Stefan Langer. 2021. Data augmentation for low-resource named entity recognition using backtranslation. *ArXiv preprint*, abs/2108.11703.

Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. 2022. Zerogen: Efficient zero-shot learning via dataset generation. *ArXiv*, abs/2202.07922.

Tongtao Zhang, Heng Ji, and Avirup Sil. 2019. Joint entity and event extraction with generative adversarial imitation learning. *Data Intelligence*, 1(2):99–120.

Zijian Zhao, Su Zhu, and Kai Yu. 2019. Data augmentation with atomic templates for spoken language understanding. 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 3637–3643, Hong Kong, China. Association for Computational Linguistics.

A Affinity and Diversity

Inspired by Gontijo-Lopes et al. (2020) and Arora et al. (2021), we proposed to use a calibration method to quantify how augmentation shifts data. They all note that a trained model is often sensitive to the distribution of the training data.

Given the original example $\boldsymbol{x}$ and one of its augmented example $\boldsymbol{x}^+$, we measure distribution shifts by computing the difference between the loss of a model trained on the original training set and tested on the original example, and the loss of the same model tested on an augmented example:

$$\tau_\alpha = \ell(M, \boldsymbol{x}) - \ell(M, \boldsymbol{x}^+), \qquad (1)$$

where M is an EE model trained on the original training set and $\ell(M, \boldsymbol{x}^+)$ denotes the model's validation loss when evaluated on the augmented example $\boldsymbol{y}$.

We use the Dist-1/2 metric (Celikyilmaz et al., 2020), commonly used in text generation, to assess the *Diversity* of the augmented data.

B Implementation Details

Parameter	Value
Training Epochs	3
Optimizer	AdamW
Batch Size	64
Learning rate	1e-5
Seed	1024
Top-k	100
Top-p	0.7
Beam Size	5

Table 3: Implementation details of our infilling model (t5-small).

Parameter	Value
Training Epochs	30
Optimizer	AdamW
Batch Size	64
Learning rate	5e-5
Seed	1024

Table 4: Implementation details of Text2Event.

Parameter	**Value**
Training Epochs	30
Optimizer	AdamW
Batch Size	64
Learning rate	4e-5
Seed	1024
nth query	5

Table 5: Implementation details of EEQA.