Leveraging Expert Guided Adversarial Augmentation For Improving Generalization in Named Entity Recognition

Aaron Reich[1,2], **Jiaao Chen**[1], **Aastha Agrawal**[1], **Yanzhe Zhang**[1], **Diyi Yang**[1]
[1]Georgia Institute of Technology
[2]Pionetechs, Inc.
{areich8, jchen896, aagrawal319, z_yanzhe, dyang888}@gatech.edu

Abstract

Named Entity Recognition (NER) systems often demonstrate great performance on in-distribution data, but perform poorly on examples drawn from a shifted distribution. One way to evaluate the generalization ability of NER models is to use adversarial examples, on which the specific variations associated with named entities are rarely considered. To this end, we propose leveraging expert-guided heuristics to change the entity tokens and their surrounding contexts thereby altering their entity types as adversarial attacks. Using expert-guided heuristics, we augmented the CoNLL 2003 test set and manually annotated it to construct a high-quality challenging set. We found that state-of-the-art NER systems trained on CoNLL 2003 training data drop performance dramatically on our challenging set. By training on adversarial augmented training examples and using mixup for regularization, we were able to significantly improve the performance on the challenging set as well as improve out-of-domain generalization which we evaluated by using OntoNotes data. We have publicly released our dataset and code at https://github.com/GT-SALT/Guided-Adversarial-Augmentation.

1 Introduction

Deep learning models have achieved great performance on many natural language processing (NLP) problems (Bahdanau et al., 2016; Devlin et al., 2019). However, many recent works have shown that these models often rely on *spurious correlations* which are not necessarily the *causal artifacts*. Thus, these models perform well on the in-distribution test set but are likely to exhibit a huge performance decline on out-of-distribution data (e.g. real world data) (Tu et al., 2020; Kaushik and Lipton, 2018; Poliak et al., 2018; Gururangan et al., 2018; Zhang et al., 2019; Glockner et al., 2018). Prior works have constructed adversarial examples for benchmarking the generalization ability of state-of-the-art NLP models on out-of-distribution examples (Kaushik et al., 2020; Zhang et al., 2019; Glockner et al., 2018). Proposed approaches such as random word swapping (Jin et al., 2020) and the appending of a sentence to the end of text (Jia and Liang, 2017) do not take into consideration the unique linguistic properties and variations associated with named entities. As a key problem setting involving the classification of semantic categories of entities (e.g., Organizations, Locations) (Nadeau and Sekine, 2007), NER is still in need of improved benchmarks of true generalization.

Previous works (Bernier-Colborne and Langlais, 2020; Fu et al., 2020; Stanislawek et al., 2019) have shown that words which have different entity labels in different scenarios often lead to frequently occurring errors of NER models. This can be especially problematic in specific domain applications where this challenging case is common. For example, when training an NER model for political text mining, it would be of great importance to differentiate between the categories of *Clinton* (Person) and the *Clinton Foundation* (Organization). We make use of this as the inspiration for designing expert-guided heuristic linguistic patterns for creating a high quality adversarial dataset for NER.

Leveraging such expert-guided heuristics, we propose an automated procedure for adversarial augmentation. We use this automated procedure to first generate adversarial examples from the test data. Since some of these automatically generated adversarial examples may lack quality in terms of syntax or semantics, we manually select only the examples that are of high quality for the construction of the challenging test set. The performance of state-of-the-art NER systems drops severely on this challenging test set. To alleviate this degradation, we first use the proposed heuristics to augment the training examples (without manually filtering the data for quality), which proves to be effective. We further utilize mixup (Zhang et al., 2018; Chen

Findings of the Association for Computational Linguistics: ACL 2022, pages 1947 - 1955

et al., 2020) as a regularization technique to interpolate the representations of the original examples and the augmented examples, leading to a smoother decision boundary and improved generalization ability (Lee et al., 2020; Wang et al., 2021b).

2 Related Work

Generating Adversarial Examples Adversarial data augmentation (Chen et al., 2021) severely influences a model's predictions without changing human judgements. It is widely leveraged to test the generalization ability of models (Wang et al., 2021a). For example, Jia and Liang (2017) fools a reading comprehension system by inserting distracting sentences. Belinkov and Bisk (2018) leverages synthesized or natural typos to attack character-based translation models. However, few prior works have explored the generation of adversarial examples specifically for NER. Gui et al. (2021) performed augmentations by concatenating sentences, swapping/inserting/deleting a random character in an entity, entity swapping with Out-of-Vocabulary entities, and cross category swapping. Zeng et al. (2020) also took a random entity swapping approach but only selected entities of the same label to preserve linguistic correctness. In this work, we purposely alter the entity type by adding/deleting tokens in predefined *word phrase* sets and alter the surrounding context.

Adversarial Training and Mixup One approach for improving a model's performance on adversarial examples is to incorporate adversarial examples into its training (adversarial training, Goodfellow et al., 2014). However, this may not improve the generalization ability of the model, since the model is only learning to focus on manipulated hard examples (Lee et al., 2020). One solution is to combine mixup Zhang et al. (2018) with adversarial training (Lee et al., 2020; Wang et al., 2021b). By linearly interpolating training data and their associated labels, mixup is able to improve the classifier's generalization ability by training on these interpolated data points which helps to form a smoother decision surface. In the context of adversarial training, mixup is leveraged to form diverse adversarial examples (Wang et al., 2021b) and prevent overfitting on adversarial features (Lee et al., 2020), thus improving the overall generalization ability. In this work, we use mixup to interpolate the original examples and expert-guided adversarial examples to improve the generalization ability of NER models.

3 Expert-Guided Adversary Generation

Current NER models often deal with unambiguous cases where one entity often gets assigned to the same label. By inducing challenging cases using the Overlapping Categories (Fu et al., 2020) that alter the entity and its label, models can then be tested to see whether they are only learning spurious correlations between the token and the label. For the construction of adversarial examples by the altering of entity types, we define three components: (i) **Eligibility Check**: We only augment entities that are eligible to change their entity types. (ii) **Entity Token Change**: By adding or deleting certain predefined tokens, we change the entity type of the original tokens to a target type. (iii) **Entity Context Change**: To deal with ambiguous tokens, we further add some predefined contexts that correspond to the target entity type. Note that predefined words/phrases/contexts used in different scenarios form different predefined *word phrase* sets, into which embed expert knowledge. During the automatic generation process, we randomly sample from the corresponding *word phrase* sets. Table 1 contains examples of expert-guided adversarial augmentations. The three components are defined below for their use in the transition to each target entity type (organization, person, location):

Organization For transitioning to ORGANIZATION, an example is considered **eligible** if an entity only contains one token (e.g. "*Brazil*"). **Entity Token Change** in this case refers to inserting words and phrases which are often used behind or after some tokens to form an organization (e.g. add "*University*" after "*Brazil*"). Such words and phrases form a set of size 44, including "*University of*" (inserted before) and "*Department*" (inserted after). **Entity Context Change** for ORGANIZATION involves inserting a suitable context after the newly formed organization entity, such as "*and its team*" and "*'s office*". Such phrases form a set of size 42.

Location Different from transitioning to ORGANIZATION, we want to instead ensure the augmented entity of type LOCATION is a real world location. To achieve this, we **combine** the eligibility check and entity token change: we first define a *word phrase* set containing words and phrases that are likely to form an organization when concatenated to a location, such as "*Bank of*" (be-

Transition	Count	Examples
Location or Person → Organization	510	**Original:** Every year, 500 new plastic surgeons graduate in Brazil and medical students from all over the world come to study there. **Augmented:**Every year, 500 new plastic surgeons graduate from Brazil University and medical students from all over the world come to study there.
Organization → Location	99	**Original:** Munich Re says to split stock. **Augmented:** Munich's largest corporation says to split stock.
Organization or Location → Person	391	**Original:** The Colts won despite the absence of injured starting defensive tackle Tony Siragusa, cornerback Ray Buchanan and linebacker Quentin Coryatt. **Augmented:** Colts Zardari and her team won despite the absence of injured starting defensive tackle Tony Siragusa, cornerback Ray Buchanan and linebacker Quentin Coryatt.

Table 1: Expert-guided transition types for producing adversarial augmentations for NER. The original entity is colored in blue and entity token change is colored in red. The entity context change is colored in brown. Note that the entity context change is not always applied in the transition to ORGANIZATION. We also provided the statistics of the challenging set.

fore America). Such phrases form a set of size 82. We then perform eligibility check by locating those organization entities containing one of such phrases and change their entity type by deleting those phrases (e.g. delete "*Re*" from "*Munich Re*"). **Entity Context Change** involves the insertion of a natural context after the entity, such as "*'s largest corporation*" and "*'s football club*". We have 16 of such contexts.

Person Similar to transitioning to ORGANIZATION, an example is considered **eligible** for transitioning to PERSON if an entity only contains one token (e.g. "*Colts*"). **Entity Token Change** in this situation refers to the insertion of a token representing a person's last name after the original token to change the entity type to PERSON (e.g. add "*Zardari*" after "*Colts*"). Such predefined tokens for insertion form a set of size 152, including examples such as "*Dutra*" and "*Martin*". **Entity Context Change** for a person then involves inserting a suitable context after the newly formed entity, such as "*and her team*" and "*and his company*". Such phrases form a set of size 49.

We include more examples of word phrases in the Appendix (Table 4) and the GitHub repository contains the full sets. Note that the automatically augmented adversarial examples may lack semantic and syntactic quality. For example, there may be grammatical issues or the randomly inserted contexts may be in conflict with current contexts. Thus we only use them for adversarial training (Section 4). To build the challenging test set, we manually select the high quality examples from the augmented test dataset (Section 5.1).

4 Mixup with Adversarial Examples

Adversarial training improves a model's robustness to adversarial examples by directly training on adversarial examples, however, such training might hurt generalization (Raghunathan et al., 2019) or cause overfitting on adversarial features (Lee et al., 2020) (predefined *word phrases* in our case). To this end, we leverage mixup (Zhang et al., 2018; Verma et al., 2019) to mitigate these issues and further improve generalization on the basis of adversarial training (Lee et al., 2020).

Given a pair of data points (x, y) and (x', y'), where x denotes a data point and y denotes its label in a one-hot representation, mixup (Zhang et al., 2018) creates a new data point by the interpolation of the data and their labels as shown below with λ being drawn from a beta distribution:

$$\hat{x} = \lambda x + (1 - \lambda)x' \quad (1)$$

$$\hat{y} = \lambda y + (1 - \lambda)y' \quad (2)$$

In this work, (x, y) is a training example that is eligible for heuristic augmentation and is paired with its heuristically modified version (x', y'). Since textual data is discrete and cannot be mixed in the input space, the interpolation of the two examples is computed in the hidden space.

Following Chen et al. (2020), Let $\mathbf{h}^m = \{h_1..h_n\}$ be the hidden representations after the m-th layer where they are the concatenation of the token representations. The hidden representation for each token in the original example at the m-th layer $\mathbf{h}^m$ is linearly interpolated with $\mathbf{h}^{m\prime}$, the representation for each token in the augmented example, by a ratio λ:

$$\hat{\mathbf{h}}^m = \lambda \mathbf{h}^m + (1 - \lambda)\mathbf{h}^{m\prime} \quad (3)$$

Then $\hat{\mathbf{h}}^m$ is passed to the $(m + 1)$-th layer, and the labels for the final output logits are mixed at the same ratio. m is randomly sampled from $\{8, 9, 10\}$. The mixing parameter λ is sampled from a beta distribution: $\lambda \sim B(\alpha, \beta)$, where α and β determine

the skew of the beta distribution. In this work, we use two different beta distributions from which to sample λ. For each pair of data points, two mixed data points are generated. One data point is closer to the original examples and the other is closer to the adversarial examples. See Appendix B for more details.

5 Experiments

5.1 Datasets and Pre-processing

In-Distribution dataset (ID) We use CoNLL 2003 (Tjong Kim Sang and De Meulder, 2003) with the BIO labeling scheme following Chen et al. (2020). In order to make mixup possible in recent transformer based models like BERT, we assigned labels to the special tokens [SEP], [CLS], and [PAD]. All models are trained on the ID training set by default. We report the results on the ID test set in the third column of Table 2.

Challenge Set (CS) For the challenging set, two graduate students who have linguistic backgrounds and are familiar with NER tasks, manually constructed the dataset consisting of the ID test set transformed by the expert-guided augmentations. The goal was to build a challenging test set containing only high quality data points, by manually labeling the quality (as high or low) and making small corrections. Before annotating the full set of augmented data, they did a test annotation of a sample size of 50 examples to calculate the annotator agreement and the resulting annotator agreement was 78%. They then manually annotated the full augmented test set which resulted in a challenging set of 1000 high quality data points.

Out-of-Domain (OOD) In addition to training on an ID training set and testing on an ID test set and challenging set, we further test the few-shot generalization ability of our proposed approach on an out-of-domain dataset: OntoNotes (Ralph Weischedel and Xue., 2011). In this setting, all models are given 5 training examples of each class from the OntoNotes (Ralph Weischedel and Xue., 2011) training set (along with the ID training data). After training, we tested their out-of-domain generalization by using an OOD test set consisting of 50 examples from the OntoNotes test set. All data points had to follow the condition that the percentage of entity tokens out of all tokens is greater than 49%. This condition serves the purpose of allowing for the evaluation of the model's performance upon mostly entity tokens. Note that OntoNotes has a more fine-grained entity category than CoNLL 2003, so we mapped the OntoNotes labels to the CoNLL 2003 labels so that the data would be compatible with our models.

5.2 Baselines and Model Settings

We train six types of models: (1) a BERT Base (Devlin et al., 2019) model on only the original training examples (*BERT*); (2) a BERT Base model on the original training examples and training examples that are augmented with the expert-guided adversarial heuristics (*BERT+AT*); (3) a *BERT+AT* model with dropout probability of 0.5 (Hinton et al., 2012) (*BERT + AT + Dropout*); (4) a BERT Base model utilizing Token-Aware Virtual Adversarial Training (TAVAT, Li and Qiu, 2020), a gradient-based adversarial training technique (*BERT + TAVAT*); (5) a BERT Base model trained with the text-based adversarial attacks proposed in Gui et al. (2021) utilizing their defined NER transformations (Appendix C) (*BERT + TextFlint*); (6) a BERT Base model utilizing mixup to linearly interpolate the original training examples with the expert-guided adversarial examples (*BERT + AT + Mixup*). Note that models using mixup are not trained on more data points, since two mixed data points are generated given a pair of data points (see Section 4).

In order to test the generalization ability of the models using the proposed adversarial augmentation, we varied the percentage of adversarial augmented examples (10%, 30%, 50%, and 100% of the total number of eligible examples) used for both the proposed adversarial training and TextFlint (Gui et al., 2021). We also used smaller predefined *word phrase* sets to augment the training data by excluding 25% of the total word phrases used in the construction of the CS.

5.3 Results and Analysis

CS As shown in Table 2, *BERT* had a significant performance decline when tested on the CS, and the prior adversarial training approach failed to increase the performance on CS, demonstrating the novel challenge proposed. Not surprisingly, *BERT+AT* can dramatically improve the model's performance on the CS, even when only 10% of the eligible augmentation is used. Incorporating mixup can consistently improve it as demonstrated on CS. While prior adversarial training severely hurt the model's performance on ID, *BERT+AT+Mixup* almost maintained its ID performance which sug-

Percent	Model	ID	CS	OOD
N/A	BERT	90.82	71.80	58.72
N/A	BERT + TAVAT	91.82	70.14	-
10%	BERT + AT	90.37	86.16	61.09
	BERT + AT + Dropout	90.1	84.97	61.86
	BERT + AT + Mixup	**90.79**	**88.79**	**67.47**
	BERT + TextFlint	88.85	54.04	66.67
30%	BERT + AT	90.84	86.42	60.76
	BERT + AT + Dropout	**90.93**	86.91	61.6
	BERT + AT + Mixup	90.85	**87.30**	**69.46**
	BERT + TextFlint	89.71	60.32	65.88
50%	BERT + AT	90.85	87.50	62.18
	BERT + AT + Dropout	90.19	**88.88**	60.83
	BERT + AT + Mixup	**90.92**	88.00	**67.47**
	BERT + TextFlint	89.55	53.49	65.48
100%	BERT + AT	90.52	87.74	57.76
	BERT + AT + Dropout	90.16	88.45	60.25
	BERT + AT + Mixup	**90.53**	**90.21**	67.07
	BERT + TextFlint	87.31	59.12	**69.05**

Table 2: F1 Scores on the original CoNLL 2003 Test Set (ID), proposed Challenging Set (CS), and Out of Domain Test Set (OOD). All the results were averaged over 3 runs. '-' refers to unstable training which causes the model to collapse. Note that in the third and fourth columns, models are trained on CoNLL 2003 training data (and their augmented versions if adversarial training is available). In the fifth column, models are trained on CoNLL 2003 training data and 5-shot examples from the OntoNotes training data (and their augmented versions if adversarial training is available).

gests the good generalization ability training with the proposed adversarial augmentation provides.

For an ablation study, we conducted experiments in which we used mixup to interpolate pairs of ID training data points, and observed a big performance gap when compared to our approach (see Figure 1 in Appendix). This proved the strategic design of mixing original examples and their expert-guided adversarial versions.

OOD In the few-shot generalization experiments, while the original *BERT* demonstrated poor performance on OOD, TextFlint significantly increased performance. *BERT + AT* only marginally outperforms *BERT* when limited examples are augmented, probably suggesting that the lack of generalization is due to naive adversarial training on the proposed augmentation. However, *BERT+AT+Mixup* significantly increased the performance as demonstrated by achieving the best performance (69.46), while also outperforming the baselines in most settings. Other than the learning of smoother decision boundaries, we also hypothesize that the interpolated representations enhance the quality of the adversarial examples' representations, thus resulting in improved generalization. This hypothesis is based on the fact that the quality of the augmented examples is sometimes limited. So the interpolation with the original data in the hidden space may help to improve the quality.

6 Conclusion

This work proposed an expert-guided adversarial augmentation for NER consisting of the altering of entity types by strategic selection and modification of tokens and their contexts. Using this augmentation strategy on CoNLL 2003 and manually filtering the generated examples for quality, we constructed a high-quality challenging test set for the NER task. We show that SOTA NER systems suffer from dramatic performance drop when evaluated on our challenging set. Beyond simply using the proposed augmentation for adversarial training, we demonstrated that leveraging mixup between original examples and their augmented versions can outperform state-of-the-art baselines on in-distribution data, the challenging set, and few-shot generalization to out-of-domain data.

Acknowledgment

We would like to thank the anonymous reviewers for their helpful comments, and the members of the Georgia Tech SALT lab for their feedback.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2016. Neural machine translation by jointly learning to align and translate.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *International Conference on Learning Representations*.

Gabriel Bernier-Colborne and Phillippe Langlais. 2020. HardEval: Focusing on challenging tokens to assess robustness of NER. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1704–1711, Marseille, France. European Language Resources Association.

Jiaao Chen, Derek Tam, Colin Raffel, Mohit Bansal, and Diyi Yang. 2021. An empirical survey of data augmentation for limited data learning in nlp.

Jiaao Chen, Zhenghui Wang, Ran Tian, Zichao Yang, and Diyi Yang. 2020. Local additivity based data augmentation for semi-supervised ner.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jinlan Fu, Pengfei Liu, Qi Zhang, and Xuanjing Huang. 2020. Rethinking generalization of neural models: A named entity recognition case study.

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking nli systems with sentences that require simple lexical inferences.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples.

Tao Gui, Xiao Wang, Qi Zhang, Qin Liu, Yicheng Zou, Xin Zhou, Rui Zheng, Chong Zhang, Qinzhuo Wu, Jiacheng Ye, Zexiong Pang, Yongxin Zhang, Zhengyan Li, Ruotian Ma, Zichu Fei, Ruijian Cai, Jun Zhao, Xinwu Hu, Zhiheng Yan, Yiding Tan, Yuan Hu, Qiyuan Bian, Zhihua Liu, Bolin Zhu, Shan Qin, Xiaoyu Xing, Jinlan Fu, Yue Zhang, Minlong Peng, Xiaoqing Zheng, Yaqian Zhou, Zhongyu Wei, Xipeng Qiu, and Xuanjing Huang. 2021. Textflint: Unified multilingual robustness evaluation toolkit for natural language processing.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment.

Divyansh Kaushik, Eduard Hovy, and Zachary C. Lipton. 2020. Learning the difference that makes a difference with counterfactually-augmented data.

Divyansh Kaushik and Zachary C. Lipton. 2018. How much reading does reading comprehension require? a critical investigation of popular benchmarks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5010–5015, Brussels, Belgium. Association for Computational Linguistics.

Saehyung Lee, Hyungyu Lee, and Sungroh Yoon. 2020. Adversarial vertex mixup: Toward better adversarially robust generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 272–281.

Linyang Li and Xipeng Qiu. 2020. Tavat: Token-aware virtual adversarial training for language understanding.

David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticæ Investigationes*, 30(1):3–26.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference.

Aditi Raghunathan, Sang Michael Xie, Fanny Yang, John C. Duchi, and Percy Liang. 2019. Adversarial training can hurt generalization.

Mitchell Marcus Martha Palmer Robert Belvin Sameer Pradhan Lance Ramshaw Ralph Weischedel, Eduard Hovy and Nianwen Xue. 2011. Ontonotes: A large training corpus for enhanced processing. In *Handbook of Natural Language Processing and Machine Translation: DARPA Global Autonomous Language Exploitation. Springer*.

Tomasz Stanislawek, Anna Wróblewska, Alicja Wójcicka, Daniel Ziembicki, and Przemyslaw Biecek. 2019. Named entity recognition - is there a glass ceiling? In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 624–633, Hong Kong, China. Association for Computational Linguistics.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Lifu Tu, Garima Lalwani, Spandana Gella, and He He. 2020. An empirical study on robustness to spurious correlations using pre-trained language models. *CoRR*, abs/2007.06778.

Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, Aaron Courville, David Lopez-Paz, and Yoshua Bengio. 2019. Manifold mixup: Better representations by interpolating hidden states.

Boxin Wang, Chejian Xu, Shuohang Wang, Zhe Gan, Yu Cheng, Jianfeng Gao, Ahmed Hassan Awadallah, and Bo Li. 2021a. Adversarial GLUE: A multi-task benchmark for robustness evaluation of language models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Haotao Wang, Chaowei Xiao, Jean Kossaifi, Zhiding Yu, Anima Anandkumar, and Zhangyang Wang. 2021b. Augmax: Adversarial composition of random augmentations for robust training. *Advances in Neural Information Processing Systems*, 34.

Xiangji Zeng, Yunliang Li, Yuchen Zhai, and Yin Zhang. 2020. Counterfactual generator: A weakly-supervised method for named entity recognition. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7270–7280, Online. Association for Computational Linguistics.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond empirical risk minimization.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: paraphrase adversaries from word scrambling. *CoRR*, abs/1904.01130.

A Expert-Guided Augmentation's Adversarial Properties

When the expert-guided augmentation is applied to an example, the entity's new label is now the ground truth label. If the model classifies based upon the spurious correlation between the remnants of the original entity and context with the original label within the newly augmented text, it will be provoking the wrong classification by the prediction of the old label. This demonstrates the augmented example's adversarial properties.

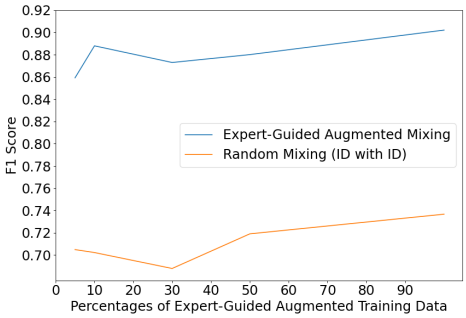

Figure 1: Random Mixing of **ID data** with **ID data** vs. Mixing of **ID data** with **Expert-Guided Augmented data**; Performances are on the CS

B Mixup Implementation Details and Hyperparameter Tuning

After sampling a λ from the beta distribution, we modify it by applying $\lambda = max(\lambda, 1-\lambda)$, which guarantees that the λ to be used is no less than 0.5. A large λ can guarantee that the resulting mixed data point ($\hat{x} = \lambda x + (1-\lambda)x'$) is always closer to x. We use two different beta distributions to sample the mixing parameter from, one for when the original examples are to be mixed (original examples as x, augmented examples as x') and one for when the heuristically augmented examples are to be mixed (augmented examples as x, original examples as x').

For the two hyperparameters corresponding to each of the two beta distributions from which the mixing parameter is sampled, α and β, we first set them at 200 and 5 respectively. We experimented with lessening the skew of the beta distribution decreasing α to 150 and while keeping β at 5. We then further experimented with increasing its skew by decreasing α to 130 and while at the same time increasing β to values of 7 and 9.

In the few-shot generalization experiments, our implementation of mixup uses four different beta distributions from which to sample the mixing parameter: Similarly, two for the in-distribution original and augmented training examples, and two for the out-of-domain original and augmented training examples.

C TextFlint NER Task Specific Transformations

The four TextFlint NER task specific transformations used are ConcatSent, EntTypos, CrossCategory, and SwapLonger. ConcatSent involves the

Percent	Model	Challenge Set
10%	BERT + AT	88.53
	BERT + AT + Dropout	83.98
	BERT + AT + Mixup	88.54
30%	BERT + AT	91.16
	BERT + AT + Dropout	93.08
	BERT + AT + Mixup	93.09
50%	BERT + AT	88.74
	BERT + AT + Dropout	93.38
	BERT + AT + Mixup	92.48
100%	BERT + AT	92.97
	BERT + AT + Dropout	93.77
	BERT + AT + Mixup	92.33

Table 3: F1 scores on the challenging set when no *word phrases* were held out during training; All of the results were averaged over 3 runs.

concatenation of two sentences into a longer one. EntTypos involves the swapping/deleting/adding of a random character to entities. CrossCategory involves the swapping of entities with ones that can be labeled by different labels. SwapLonger involves the substituting of the short entities for longer ones. Since only ConcatSent and EntTypos were available through the TextFlint framework during the time of this work, we reimplemented CrossCategory and SwapLonger for the experiments.

D No Word Phrases Held Out Experiments

In Table 3, we provide the results when using all of the *word phrases* for adversarial augmentation during training. Compared to the setting where 25% of the word phrases were held out for training (Table 2), the models experienced a significant drop in performance. The models may have learned the spurious correlation between the words from the *word phrase* set and the entity labels instead of learning the linguistic relation. This demonstrates that even though BERT's performance increases when trained on the expert-guided augmented data, the challenging set is still not "solved" as the removal of 25% of the word phrases from training caused this significant of a performance drop. This "held out" setting simulates the real world deployment of NER models.

E Tuning of TAVAT's Hyperparameters

The hyperparameters unique to Token-Aware Virtual Adversarial Training (TAVAT) such as the adversarial training step, the constraint bound of the pertubation, the adversarial step size, and the initialization bound are tuned using the values in Li and Qiu (2020).

F Experimental Details:

F.1 Description of computing infrastructure used:

GEFORCE RTX 2080 CUDA Version: 11.0

F.2 Runtime

- Training: 2 to 2 and 1/2 hours.
- Inference: 3 minutes or less

F.3 Parameters

BERT contains 110 million parameters.

F.4 Hyperparameters for Training without 5-Shot

- BERT: max sequence length 256, batch size 8, number of training epochs 10, adam epsilon=1e-08, learning rate=5e-05, weight decay=0.0
- All dropout models have dropout probability set to 0.5 for all fully connected layers in the embeddings, encoder, and pooler.
- Mixup 10 % Augmented data:
  - Original examples: α=130 β=9
  - Augmented examples: α=200 β=5
- Mixup 30 % Augmented data:

Target Entity	Word Phrase Set	Examples
Organization	Entity Token Change	Department of Transportation | Reserve Bank of | Workers Party | Corporation
	Entity Context Change	, and its ministers, | 's star player | and its services | with its government officials
Location	Entity Token Change	Court of Appeals | Stock Exchange | UNITED | Radio
	Entity Context Change	's leading newsroom | 's countryside | 's hockey team
Person	Entity Token Change	Doorn | Liano | Bronckhorst | Aynaoui | Goey | Sidhu | Bedie
	Entity Context Change	's company | and other politicians | , an accomplished player

Table 4: More examples from the predefined *word phrase* sets ; A vertical bar (|) is used to separate word phrases.

  - Original examples: α=150 β=5
  - Augmented examples: α=200 β=5

- Mixup 50 % Augmented data:
  - Original examples: α=130 β=7
  - Augmented examples: α=200 β=5

- Mixup 100 % Augmented data:
  - Original examples: α=150 β=5
  - Augmented examples: α=200 β=5

- TAVAT Model: adv init mag=0.2, adv lr=0.05, adv max norm=0.5, adv steps=2, adv train=1

F.5 Hyperparameters for 5-Shot Training

- Mixup 10 % Augmented data:
  - Original examples: α=150 β=5
  - Augmented examples: α=200 β=5
  - Original OOD examples: α=200 β=5
  - Augmented OOD examples: α=130 β=7

- Mixup 30 % Augmented data:
  - Original examples: α=200 β=5
  - Augmented examples: α=150 β=5
  - Original OOD examples: α=200 β=5
  - Augmented OOD examples: α=130 β=7

- Mixup 50 % Augmented data:
  - Original examples: α=150 β=5
  - Augmented examples: α=200 β=5
  - Original OOD examples: α=200 β=5
  - Augmented OOD examples: α=130 β=7

- Mixup 100 % Augmented data:
  - Original examples: α=130 β=5
  - Augmented examples: α=200 β=5
  - Original OOD examples: α=200 β=5
  - Augmented OOD examples: α=130 β=7

- TAVAT Model, 5-Shot Training: adv init mag=0.2, adv lr=0.05, adv max norm=0.5, adv steps=2, adv train=1

F.6 Dataset

- CoNLL 2003 Language: English
- Training set for CoNLL 2003: Number of examples: 14041
- Dev set for CoNLL 2003: Number of examples: 3250
- Test set for CoNLL 2003: Number of examples: 3453