How Effective and Robust is Sentence-Level Data Augmentation for Named Entity Recognition?

Runmin Jiang[1,2], Xin Zhang[3], Jiyue Jiang[4], Wei Li[1(✉)], and Yuhao Wang[1,2(✉)]

[1] School of Information and Engineering, Nanchang University, Jiangxi, China
rmjiang@email.ncu.edu.cn, {liwei.cs,wangyuhao}@ncu.edu.cn
[2] Industrial Institute of Artificial Intelligence, Nanchang University, Jiangxi, China
[3] College of Computer and Cyber Security, Fujian Normal University, Fujian, China
[4] Department of Computer Science, The University of Hong Kong, Hong Kong SAR, China
jiangjy@connect.hku.hk

Abstract. Data augmentation is a simple but effective way to improve the effectiveness and the robustness of pre-trained models. However, they are difficult to adapt to token-level tasks such as named entity recognition (NER) because of the different semantic granularity and more fine-grained labels. Inspired by some mixup augmentations in computer vision, we proposed three sentence-level data augmentations including CMix, CombiMix, TextMosaic, and adapted them to the NER task. Through empirical experiments on three authoritative datasets (OntoNotes4, CoNLL-03, OntoNotes5), we found that these methods will improve the effectiveness of the models if controlling the number of augmented samples. Strikingly, the results show our approaches can greatly improve the robustness of the pre-trained model even over strong baselines and token-level data augmentations. We achieved state-of-the-art (SOTA) in the robustness evaluation of the CCIR CUP 2021. The code is available at https://github.com/jrmjrm01/SenDA4NER-NLPCC2022.

Keywords: Sentence-level data augmentation · Named entity recognition · Effectiveness · Robustness

1 Introduction

As a classic research topic, Named Entity Recognition (NER) is commonly adopted in the field of Natural Language Processing (NLP) [1]. It is known to all that the high performance of the NER task depends on the size and quality of the effective and robust pre-trained model [2]. At the same time, NER models have seen significant improvement in their performance with the recent advances of pre-trained language models [3], yet obtaining massive and diverse labeled data is often expensive and time-consuming [4]. Even if a large annotated corpus has already been obtained beforehand, it will inevitably have rare entities that do not appear enough to train the model to recognize them accurately in the text [5]. Therefore, the data augmentation method for NER is crucially significant [6].

W. Lu et al. (Eds.): NLPCC 2022, LNAI 13551, pp. 55–66, 2022.
https://doi.org/10.1007/978-3-031-17120-8_5

Previous work has studied lot of data augmentations for sentence-level tasks such as text classification.[4, 7–9, 21] However, because of the different semantic granularity and more fine-grained labels, they are difficult to adapt to tasks for token-level classification such as named entity recognition. Besides, the big models are often brittle to adversarial examples because of the overfitting, resulting in bad robustness for the NER tasks. [10] Dai & Adel [11] implemented research that mainly paid their attention to the simple data augmentation methods and adapted them to NER, but lack of research on data augmetation at the sentence level. Some studies have explored the impact of mixup augmentation on robustness evaluation, but lack of more variant methods and its impact on effectiveness for pre-trained model [10, 12].

To facilitate research in this direction, inspired by the Cutmix [13], Augmix [14], and Mosaic [15] augmentation methods in computer vision, we proposed three sentence-level data augmentation methods for named entity recognition including CMix, CombiMix, TextMosaic. We conducted empirical experiments on three authoritative datasets comparing our proposed method with a strong baseline (no data augmentations) and mentioned replacement (MR) which is one of the representative token-level data augmentations [11]. We find that the data augmentation methods may not necessarily improve the effectiveness of the models. However, our proposed methods are always better than the token-level method both in effectiveness and robustness. If controlling the number of augmented samples, these methods will enhance the performance of the pre-trained models. The results also show that our approaches can greatly improve the robustness of the pre-trained model even over strong baselines, and we achieved SOTA in the robustness evaluation of the CCIR CUP 2021. We release our code at https://github.com/jrmjrm01/SenDA4NER-NLPCC2022.

2 Methodology

2.1 CMix

The core idea of CMix method is that we need to randomly select a group of data from the replacement sentence source and randomly replace any group of data from the target sentence source. Before using the CMix method, there are two sentence sources, one is the target sentence source, and the other is the replacement sentence source. When randomly cutting the data from the replacement sentence source, the data from the replacement sentence source is replaced with the data from the target sentence source in a random mixing ratio of 0% to 50%, and so on for each target sentence source. However, at most 50% of the data in the target sentence source will be randomly replaced with the data in the replacement sentence source.

Algorithm1 CMix

```
1:Input: The original texts and tags
2: function Cmix(sentences, tags):
3:     for pair = 0 to pair < len(sentences) by pair++ do
4:         choose the reasonable data and range randomly
5:         calculate the values new_sent and new_tag for this round of replacement
6:     end for
7:     extend sentences and tags to the end of empty lists new_sents and new_tags
8:     mess up pair of new_sents and new_tags randomly
9: return new_sents, new_tags
10: end function
11: Output: The scrambled pairs of new_sents and new_tags values
```

2.2 CombiMix

The core idea of CombiMix method is to perform different data argument methods on samples and fuse them so as to achieve the effect of data argument. This approach also requires two sentence sources, the target sentence source and the replacement sentence source. CombiMix mainly applies to two data argument methods [11], mention replacement (MR) and label-wise Token replacement (LwTR). MR uses the binomial distribution to decide whether to replace the mentions of the target sentence source. If replacement is required, the mentions in the replacement sentence source are used to replace the mentions of the target sentence source and required to be of the same label. LwTR uses binomial distribution to determine whether each word of the target sentence source is replaced or not. If replacement is needed, a word with the same label in the replacement sentence source is randomly selected for replacement and the original label sequence remains unchanged. Finally, we fuse the data set processed by MR, and the data set processed by LwTR and the original data set form the final data set of CombiMix.

Algorithm2 CombiMix

```
1: Input: The original texts, tags and tag_scheme; ratios MR_ratio and LwTR_ratio
2: function CombiMix(sentences, tags, tagScheme, MR_ratio, LwTR_ratio):
3:     convert tags from sequences to spans
4:     while sp_id < len(sentences) do
5:         replace the entity, text, tag according to MR_ratio
6:         extend cur_sent and cur_tag to the end of new_sents and new_tags
7:     end while
8:     for pair = 0 to pair < len(sentences) by pair++ do
9:         choose texts and tags randomly according to LwTR_ratio
10:        extend lwtr_sent and lwtr_tag to the end of lwtr_sents and lwtr_tags
11:    end for
12:    extend new_sents, lwtr_sents, texts to the end of mix_sents
13:    extend new_tags. lwtr_tags, tags to the end of mix_tags
14:    return function Combimix
15: end function
16: Output: The function itself and the augment data of the CombiMix method
```

2.3 TextMosaic

In this section, the method of TextMosaic will be introduced in detail. This method consists of three approaches including ***span sampling***, ***random sampling*** and ***over-sampling*** respectively, which can use sentence-level contexts and help train a more effective and robust NER model.

Span Sampling. It is kind of method to allow training data to be sampled across one or more sentences to obtain richer training accuracy, as illustrated in Fig. 1. By randomly selecting head and sampling length, the truncated parts of one or more sentences are obtained to form a new sentence. Generally speaking, there will be a logical association between the upper and lower sentences, especially the end of the upper sentence and the beginning of the next sentence.

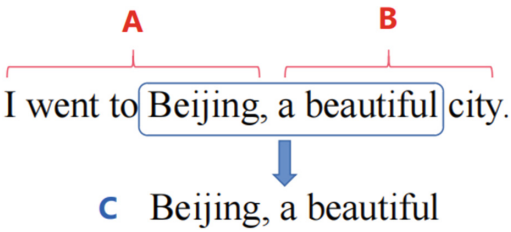

Fig. 1. The schematic diagram of span sampling. For example, sentence B is the next sentence of A, we might sample the word sequence C from the two sentences.

Random Sampling. The method refers to randomly extracting two or more data fragments from the original data and recombining them into new sentences for training, as shown in Fig. 2. By randomly sampling some fragments in different sentences, new sentences can be recombined for training. Due to certain entities can be accurately identified under common sense without much attention to contexts, thus, by superimposing multiple fragments, the diversity of textual information can be enhanced significantly. In addition, it should be noted that when randomly intercepting fragments, the interception position is generally three to five tokens before the start tag of an entity.

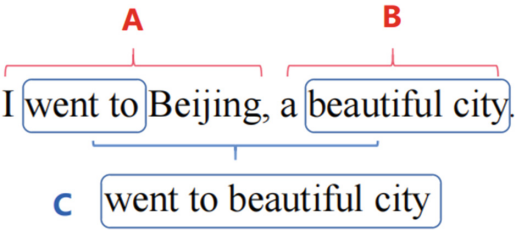

Fig. 2. The schematic diagram of random sampling. In above, the trained word sequence was sampled with two or several word sequence pieces. For example, the word sequence of C was sampled from the two sentences of A and B.

Over-Sampling. To solve the problem of uneven distribution of data labels, as shown in Fig. 3. We use the sliding window to amplify the data, which can be regarded as the process of sieving. The sliding window sampling is performed on the original context according to the specific steps. On the one hand, the position encoding of BERT is obtained by learning, so that the texts sampled by the sliding window do not overlap because of different positions. On the other hand, the specific step is obtained by sliding window sampling, which reduces the operational steps of filling and optimizes the training process of the model.

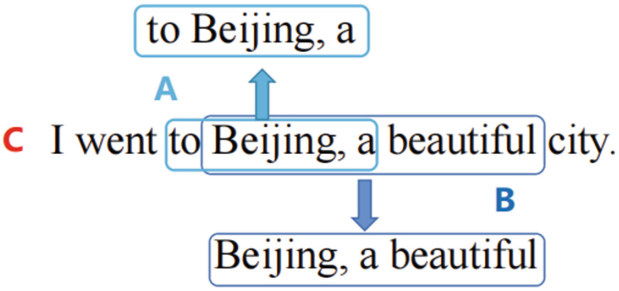

Fig. 3. The schematic diagram of over-sampling. For example, the word sequence A was sampled from the total sentence C, and with one step shifted, the word sequence B was sampled from the total sequence above.

3 Experiment

This section first introduces three authoritative NER datasets and their post-attack dataset by TextFlint [16] in Sect. 3.1, and then shows the experimental setup in Sect. 3.2. We present main results of the effectiveness evaluation in Sect. 3.3 and further explore how the augmented sample size influences the effectiveness of these methods in Sect. 3.4. The results show our method greatly improves the robustness of the pre-trained model even over strong baselines in Sect. 3.5. What's more, we participated in a NER Robustness Competition hosted by TexfFlint, where our approach achieved state-of-the-art (SOTA) in CCIR Cup2021 in Sect. 3.6.

3.1 Datasets

In order to evaluate the effectiveness of our proposed approaches, we conduct experiments on three authoritative and popular NER datasets across two languages, including the OntoNotes4.0 Chinese dataset [17], OntoNotes5.0 English dataset [18], and CoNLL-03 English dataset [19]. We show descriptive statistics of these datasets in Table 1.

In order to evaluate the effectiveness of our proposed approaches, we conduct experiments on the above datasets that were attacked by TextFlint [16]. This includes many

diverse methods of attack such as universal text transformation, adversarial attack, subpopulation and their combinations. We combined the datasets of OntoNotes5.0 and CoNLL-03 mentioned above after being attacked by 20 different attack methods[1,2], and evaluate the robustness of our proposed methodology. The OntoNotes4.0 dataset after being attacked by TextFlint as a benchmark competition for the CCIR CUP 2021[3].

Table 1. The statistics of the adopted datasets.

	OntoNotes4			CoNLL-03			OntoNotes5		
	Train	Dev	Test	Train	Dev	Test	Train	Dev	Test
Sentence	15723	4300	4345	14897	3466	3684	82727	10507	10393
Tokens	491903	200505	208066	203621	51362	46435	1299312	163104	169579
Mentions	41203	20573	22918	23499	5942	5648	41203	20023	22918
Entity Types	4	4	4	4	4	4	18	18	18

3.2 Experimental Setup

Baseline. Named entity recognition can be modeled as a sequence-label task. The state-of-the-art sequence models consist of distributed representations for input, context encoder, and tag decoder [20]. We adopt the BERT model [3] as the backbone model for pre-training and decoded by linear layers, then fine-tuned on the NER dataset, as shown in Fig. 4. For the BERT embedding, we used the following Huggingface-pretrained BERT models: "bert-base-chinese"[4] for the Chinese dataset and "bert-base-uncased"[5] for the English dataset. In the baseline, we do not use any data augmentation methods and set the same hyperparameters and pipeline for the following experiment.

Token-level Augmentation. Current token-level data augmentations dedicated to named entity recognition are label-wise token replacement (LwTR), mention replacement(MR), and synonym replacement(SR). [11] We choose one of the ***representative*** token-level methods that is ***MR*** and compare three sentence-level data augmentations with it.

Training. To improve the convergence and robustness of the model, we use a bag of tricks [21] and select the optimal hyper-parameters as shown in Table 2. The gradient accumulation method can achieve a similar effect to a large batch size when the algorithm is limited. Therefore, the learning rate warm-up method is utilized to speed up the

[1] https://www.textflint.com/.
[2] https://github.com/CuteyThyme/Robustness_experiments.
[3] https://www.datafountain.cn/competitions/510/datasets.
[4] https://huggingface.co/bert-base-chinese.
[5] https://huggingface.co/bert-base-uncased.

convergence speed. In addition, using the encapsulated optimizer AdamW [22], each parameter can be given an independent learning rate, and the past gradient history can be taken into consideration as well. To alleviate the over-fitting problem, label smoothing and limiting the non-linear parameters are conducted to solve the dilemma of over-fitting. The method of multi-model ensemble stacking uses 3-fold cross-validation.

Table 2. The hyper-parameters of the experiment

Hyperparameter	Value
Learning rate	0.00024
Weight decay	5e–3
Batch size	8
Gradient accumulation	8(step)
Warmup	5(epoch)

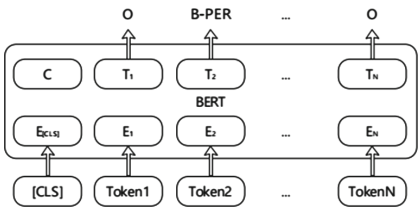

Fig. 4. The training model. BERT(backbone) + NER(head)

Metric. We adopted *span-based Micro F1* to measure the effectiveness and robustness except for CCIR Cup. The CCIR Cup used *span-based Macro F1* to evaluate the robustness.

3.3 Results of Effectiveness Evaluation

Table 3 shows the overall results of effectiveness evaluation on CoNLL-03, OntoNotes4, and OntoNotes5 datasets. However, we would also like to report a negative result, which does not apply to all datasets, such as OntoNotes4 and CoNLL-03, where the performance is reduced (except CMix) using data augmentation methods. However, at the same time, compared to MR, our proposed methods mostly outperform results, demonstrating that the sentence-level data augmentation methods are also relatively effective.

Table 3. Results of effectiveness evaluation

Dataset	Baseline	MR	CMix	CombiMix	TextMosaic
CoNLL-03	89.51	88.64	88.87	87.64	**89.07**
OntoNotes5	51.21	65.81	**67.21**	**66.18**	58.78
OntoNotes4	78.48	78.24	**78.78**	77.61	77.49

3.4 Study of the Sample Size After Data Augmentation

We counted the number of samples after data augmentation for the three training sets as shown in Table 4. MR and Cmix were twice as large as baseline, and CombiMix increased the number of samples three times as large as Cmix. The sample size for the TextMosaic(set sample length = 64) was the largest on OntoNotes4 and OntoNotes5.

Table 4. In OntoNotes4, CoNLL-03 and OntoNotes5 datasets, the data sample size after processing by five data argument methods

Datasets	Baseline	MR	CMix	CombiMix	TextMosaic
OntoNotes4	965	1930	1929	5780	7679
CoNLL-03	6893	13846	13350	39883	3175
OntoNotes5	2836	5632	5642	16938	20295

To further explore the effect of the number of samples and their distribution characteristics on the model performance after data enhancement, we take OntoNotes4 as an example and balance all samples to the same value with the following strategy.

- Baseline: Duplicate original samples three times to 3860 samples
- MR: Duplicate original samples one times to 3860 samples
- CMix: Duplicate original samples one times to 3860 samples
- CombiMix: Shuffle original samples and randomly select 3860 samples
- TextMosaic: Shuffle original samples and randomly select 3860 samples

Table 5 shows the change in F1 score before and after balancing the number of samples. We found that duplicate samples could also be used as a means of data enhancement. Although the number of duplicate samples did not change the characteristics of the data distribution, Baseline, CMix also improved by 1% compared to the previous one. In our analysis, it is possible that the data augmentation carries a large amount of irregular semantic information and noise, reducing the performance of the model. And thus the performance of the model is reduced, although it may be able to improve the robustness of the model.

Table 5. Compare the F1 score of the five data argument methods with the current F1 score before and after balancing the number of samples

	Baseline	MR	CMix	CombiMix	TextMosaic
No balanced F1	78.48	78.24	78.78	77.61	77.49
Balanced F1	79.22↑	78.24–	**79.35**↑	78.07↑	**79.37**↑

3.5 Results of Robustness Evaluation

The two datasets contain twenty transformations, such as word changes, back translations, contraction, extended sentences by irrelevant sentences, keyboard error and so on, as illustrated in Fig. 5 and Fig. 6. Table 6 shows overall results of robustness evaluation on CoNLL-03, and OntoNotes5 datasets that were attacked by TextFlint. On the one hand, the F1 of the model for both datasets dipped 7%–17% approximately. On the other hand, these methods can improve the robustness of the model, with both CombiMix and TextMosaic being higher than baseline and MR on CoNLL-03. On OntoNotes 5, all three sentence-level data augmentations show significant improvements over MR and Baseline, with the best results method CMix was 17% higher than strong baseline.

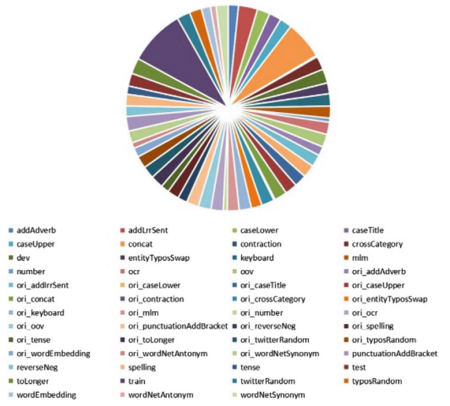

Fig. 5. CoNLL-03 data size distribution after attack by TextFlint

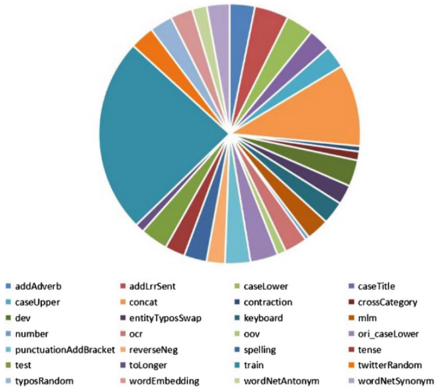

Fig. 6. OntoNotes5 data size distribution after attack by TextFlint

Table 6. Comparison of F1 score of five methods on robustness evaluation

Dataset	Baseline	MR	CMix	CombiMix	TextMosaic
CoNLL-03	83.28	82.73↓	82.74	**83.29**↑	**85.55**↑
OntoNotes5	43.38	52.96	**60.62**↑	**56.61**↑	**54.12**↑

3.6 Results of CCIR Cup

We participated in a robustness evaluation competition hosted by TextFlint in CCIR Cup 2021[6]. The validation sets and test sets used for the evaluation were generated by TextFlint after performing eleven forms of changes on OntoNotes4. The evaluation was divided into two phases, with LeaderBoard A (LB-A) focusing on contextual changes and LeaderBoard B(LB-B) combining more forms of contextual changes and entity changes. We test three proposed sentence-level augmentations and reported main results in Table 7. We achieved ***first place*** in both phases. In LB-A we got the highest F1 score with **85.99** which is 7.96 higher than the second place(F1 = 78.03), and in LB-B we got an F1 score to **76.54** which is 2.53 higher than second place(F1 = 74.01).

Different from the experiment setup, we use data augmentation methods before pre-training and semi-supervised learning in combination with out-of-domain dataset [23]. In our analysis, using generic data augmentation as a noise agent for the consistent training method may be a good choice.

Furthermore, we tested the length of the predicted sequence in model inference and found that the effect is best when the sequence length is 126. In our analysis, when the condition of the sequence length is too long, the long-distance dependence learning effect of the transformer is not good, resulting in poor model performance; Conversely, when the sequence length is too short, it is difficult to learn the semantic information of the entity context, resulting in poor NER performance.

Table 7. Results of CCIR CUP(S510/S254/S126: set sequence length = 510/254/126)

Base-line	+CMix	+CombiMix	+Text-Mosaic	S510	S254	S126	+Tricks	LB-A
▲				▲				70.14
▲			▲	▲				76.97
▲			▲		▲			80.30
▲			▲			▲		83.74
▲	▲					▲	▲	85.94
▲		▲				▲	▲	85.95
▲			▲			▲	▲	**85.99**

4 Conclusion

This research proposes three different strategies for sentence-level data augmentation for named entity recognition, which is a token-level task. Through experiments on three authoritative datasets, we find that the data augmentation methods may not necessarily

[6] https://ccir2021.dlufl.edu.cn/ccirContest/index.html.

improve the effectiveness of the models but controlling the number of augmented samples will enhances the performance of the pre-trained models to fit the feature distribution of the input contextual embeddings. The results also show that our approach can greatly improve the robustness of the pre-trained model even over strong baselines, and we achieved state-of-the-art (SOTA) in the CCIR CUP 2021.

Acknowledgement. This work was supported by the National Key Research and Development Project under Grant 2018YFB1404303, and 03 special project and 5G project of Jiangxi Province of China (Grant No.20203ABC03W08), and the National Natural Science Foundation of China under Grant 62061030 and 62106094, and the Natural Science Foundation of Zhejiang Province of China (Grant No.LQ20D010001). We would like to thank Xiangyu Shi for his contribution to the comparison experiment, and Professor Xipeng Qiu, Professor Xiangyang Xue, Professor Dongfeng Jia and Dr. Hang Yan for their guidance on this paper. Thanks to the reviewers for their hard work to help us improve the quality of this paper.

References

1. Li, J., Sun, A., Jianglei H., Li, C.: A survey on deep learning for named entity recognition. IEEE Trans. Knowl. Data Eng. **34**(1), 50-70 (2020a)
2. Wang, Y., et al.: Application of pre-training models in named entity recognition. In: 2020 12th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), vol. 1. IEEE (2020)
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL, pp. 4171–4186, Minneapolis, Minnesota (2019)
4. Wei, J, Zou, K. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. arXiv preprint arXiv:1901.11196 (2019)
5. Fritzler, A., Logacheva, V., Kretov, M.: Few-shot classification in named entity recognition task. In: Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, pp. 993-1000 (2019)
6. Feng, S.Y., Gangal, V., Wei, J., et al.: A survey of data augmentation approaches for nlp. arXiv preprint arXiv:2105.03075 (2021)
7. Karimi, A., et al.: AEDA: an easier data augmentation technique for text classification. In: EMNLP (2021)
8. Yoon, S., Kim, G., Park, K.: SSMix: Saliency-Based Span Mixup for Text Classification. ArXiv, abs/2106.08062 (2021)
9. Sun, L., et al.: Mixup-transformer: dynamic data augmentation for NLP tasks. Coling (2020)
10. Lin, B., Yuchen, J., et al.: RockNER: A simple method to create adversarial examples for evaluating the robustness of named entity recognition models. In: EMNLP (2021)
11. Dai, X., Adel, H.: An analysis of simple data augmentation for named entity recognition. ArXiv abs/2010.11683 (2020)
12. Si, C., et al.: Better robustness by more coverage: adversarial and mixup data augmentation for robust finetuning. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pp. 1569-1576 (2021)
13. Yun, S., Han, D., Oh, S.J., et al.: Cutmix: Regularization strategy to train strong classifiers with localizable features. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6023–6032 (2019)

14. Hendrycks, D., Mu, N., Cubuk, E.D., et al.: Augmix: A simple data processing method to improve robustness and uncertainty. arXiv preprint arXiv:1912.02781 (2019)
15. Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 (2020)
16. Gui, T., et al.: TextFlint: unified multilingual robustness evaluation toolkit for natural language processing. ArXiv abs/2103.11441 (2021)
17. Weischedel, R., et al.: Ontonotes release 4.0. LDC2011T03, Philadelphia, Penn Linguist. Data Consortium (2011)
18. Pradhan, S.: Towards robust linguistic analysis using ontonotes. In Proceedings of the Seventeenth Conference on Computational Natural Language Learning, CoNLL 2013, Sofia, Bulgaria, 8–9 August 2013, pp. 143–152. ACL (2013)
19. Erik, F., Sang, T.K., De Meulder, F.: 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, pp. 142–147 (2003)
20. Li, J., Sun, A., Han, J., et al.: A survey on deep learning for named entity recognition. IEEE Trans. Knowl. Data Eng. **34**(1), 50–70 (2020)
21. Sun, C., Qiu, X., Xu, Y., Huang, X.: How to fine-tune bert for text classification? In: Sun, M., Huang, X., Ji, H., Liu, Z., Liu, Y. (eds.) CCL 2019. LNCS (LNAI), vol. 11856, pp. 194–206. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32381-3_16
22. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
23. Longpre, S. et al.: How Effective is Task-Agnostic Data Augmentation for Pretrained Transformers? Findings(2020)