Improving Low-Resource Named Entity Recognition via Label-Aware Data Augmentation and Curriculum Denoising

Wenjing Zhu*, **Jian Liu***, **Jinan Xu**†, **Yufeng Chen**, **Yujie Zhang**
School of Computer and Information Technology, Beijing Jiaotong University,
Beijing 100044, China
{18120461,jianliu,jaxu,chenyf,yjzhang}@bjtu.edu.cn

Abstract

Deep neural networks have achieved state-of-the-art performances on named entity recognition (NER) with sufficient training data, while they perform poorly in low-resource scenarios due to data scarcity. To solve this problem, we propose a novel data augmentation method based on pre-trained language model (PLM) and curriculum learning strategy. Concretely, we use the PLM to generate diverse training instances through predicting different masked words and design a task-specific curriculum learning strategy to alleviate the influence of noises. We evaluate the effectiveness of our approach on three datasets: CoNLL-2003, OntoNotes5.0, and MaScip, of which the first two are simulated low-resource scenarios, and the last one is a real low-resource dataset in material science domain. Experimental results show that our method consistently outperform the baseline model. Specifically, our method achieves an absolute improvement of 3.46% F_1 score on the 1% CoNLL-2003, 2.58% on the 1% OntoNotes5.0, and 0.99% on the full of MaScip.

1 Introduction

Named entity recognition (NER) is a fundamental natural language processing (NLP) task aiming to identify the names of people, places, organizations, and proper nouns in texts, which supports a wide range of downstream applications (Huang et al., 2015; Kuru et al., 2016). The current state-of-the-art methods for NER rely on abundant training data. However, manual annotation is expensive, which limits the effectiveness of the model, especially in bio-medicine and material chemistry domains (Friedrich et al., 2020). Many studies have investigated NER in low-resource scenarios, by transferring pre-trained language representations on self-supervised or rich-resource domains to target domains (Ruder, 2019; Gururangan et al., 2020). Others use the knowledge base to semi-automatically label extra data for training (Zeng et al., 2015). Nevertheless, these methods usually require huge expertise knowledge to obtain good performance.

Data augmentation has been proven effective to alleviate data scarcity in many NLP tasks, including machine translation (Wang et al., 2018; Gao et al., 2019), text classification (Wei and Zou, 2019; Xie et al., 2020), question answering (Raiman and Miller, 2017), etc. (Min et al., 2020). However, most existing studies focus on sentence-level tasks, which generate sentences via word replacement, swap, and deletion (Wei and Zou, 2019; Min et al., 2020) or generative models (Yu et al., 2018; Iyyer et al., 2018). Different from these sentence-level NLP tasks, NER predict entities on the token level. That is, for each token in the sentence, NER models predict a label indicating whether the token belongs to a mention and which entity type the mention has. Therefore, applying transformations to tokens may also change their labels. Due to this difficulty, data augmentation for NER is comparatively less studied.

In this work, we propose a novel data augmentation framework for NER in low-resource scenarios, which generates examples with consistent labels and filters noises in the generated data. Concretely, our approach contains two complementary components: 1) *data augmentation via pre-trained BERT*, which

*Equal contribution
†Corresponding author

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1131-1142, Hohhot, China, August 13 - 15, 2021.
(c) Technical Committee on Computational Linguistics, Chinese Information Processing Society of China

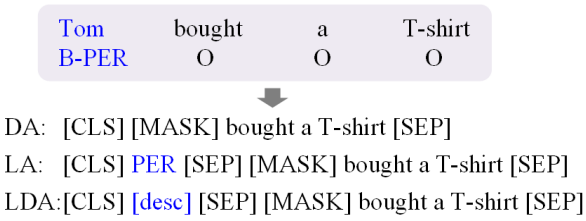

Figure 1: The input format of different data augmentation methods. DA, LA, LDA correspond to the basic method, label additional method, and label description additional method. As shown in blue, LA puts the label at the beginning of the sentence, while LDA uses the description of the label.

uses contextualized information to predict a masked word for data augmentation, and 2) *data denoising via curriculum learning*, which filter noises in the augmented data for further boost learning. In *data augmentation via pre-trained BERT*, our basic idea is to predict the masked words through pre-trained language models (we use BERT (Devlin et al., 2019) in this paper), and then replace original words with predicted words to generate new sentences. However, directly using BERT for prediction may generate some words that mismatch the original labels. As shown in Figure 1, given a sentence *"Tom bought a T-shirt"*, we replace *"Tom"* with [MASK] and predict it by BERT. The predicted words may be third-person pronouns like *"he"*, *"she"* or wrong words, which causes mismatch between original labels and generated words. In response to this issue, we propose a label-aware data augmentation method, which considers the label information to make the predicted words match the original label more closely. In *data denoising via curriculum learning*, we propose a new method based on curriculum learning to filter augmented data. Considering that the synthetic data still contain noises, we design three evaluation metrics to measure the generated examples by confidence, and then use curriculum learning strategy to filter noises. Consequently, the performance is significantly improved.

To evaluate the effectiveness of our approach, we conduct experiments in both simulated low-resource scenarios and real-world low-resource scenarios. In the former scenarios, we use two standard NER datasets: CoNLL-2003 (Tjong Kim Sang and De Meulder, 2003) and OntoNotes5.0 (Pradhan et al., 2013), which are randomly sampled to simulate a low-resource scenarios. In the latter scenarios, we use a dataset from the material science domain: MaScip (Mysore et al., 2019). The results show that our method obtains a significant performance improvement over baseline model (i.e., Bi-LSTM-CRF for NER)

Our contributions are summarized as follows:

- We propose a novel data augmentation method for low-resource NER task, which uses pre-trained BERT to generate label-aware synthetic data and curriculum learning strategy on generated data denoising to improve data quality.
- We conduct experiments on two standard NER datasets and a real-world low-resource NER dataset. Experimental results demonstrate the effectiveness of our methods in low-resource scenarios. Moreover, our methods can be easily applied to other token-level tasks.

2 Related Work

2.1 Low-resource NER

Deep learning-based methods achieve good performance for NER with abundant annotated data but encounter various challenges when the labeled data is scarce. Therefore, more and more works pay attention to improving the performance of NER in low-resource scenarios. Kruengkrai et al. (2020) use sentence-level information in auxiliary tasks to improve model performance on low-resource languages. Peng et al. (2019) use dictionaries to directly label data, ignoring entities that are not in the dictionaries. This method greatly reduce the requirements on the quality of the dictionaries. Shang et al. (2018) propose a revised fuzzy CRF layer to handle tokens with multiple possible labels and a neural model

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1131-1142, Hohhot, China, August 13 - 15, 2021.

AutoNER with a new Tie or Break scheme. Han and Eisenstein (2019) propose domain adaptive fine-tuning with unlabeled data to reduce the discrepancy between different domains. All of these methods focus on existing resources and do not consider using synthetic data for data augmentation.

Several works have studied using data augmentation for NER. Mathew et al. (2019) train a weak tagger to annotate unlabeled data through weak supervision. Dai and Adel (2020) summarize the sentence-level and sentence-pair level data augmentation methods on the NLP tasks and apply some of them to the NER task, including the token replacement, synonym replacement, mention replacement, and shuffle within segments. Synonym replacement often relies on external knowledge, e.g. WordNet (Miller, 1995), which is a manually designed dictionary that may have low coverage (or not available) for low-resource languages. Ding et al. (2020) propose an augmentation method with language models trained on the linearized labeled sentences.

Different from the above methods, our approach uses pre-trained BERT to predict the masked words, which contains rich contextual information. Then the masked words are replaced with the predicted words to generate new synthetic sentences, and the original label sequence remains unchanged.

2.2 Curriculum Learning

Curriculum learning (Bengio et al., 2009) is a particular learning paradigm in machine learning, which starts from easy instances and then gradually handles harder ones. Liu et al. (2018) propose a natural answer generation framework based on curriculum learning for question answering tasks. Pentina et al. (2015) use curriculum learning to study the best order of learning tasks in multitasking problems. Platanios et al. (2019) propose a neural machine translation framework based on curriculum learning. It decides which training samples to show to the model at different times during the training process according to the estimated difficulty of the samples and the current capabilities of the model, which greatly reduce the training time. Matiisen et al. (2020) propose Teacher-Student Curriculum Learning, a framework for automatic curriculum learning, where the Student try to learn a complex task, and the Teacher automatically choose subtasks from a given set for the Student to train on. Gong et al. (2016) employ the curriculum learning methodology by investigating the difficulty of classifying every unlabeled image. The reliability and the discriminability of these unlabeled images are particularly investigated for evaluating their difficulty. As a result, an optimized image sequence is generated during the iterative propagations, and the unlabeled images are logically classified from simple to difficult. Wang et al. (2019) propose a unified framework called Dynamic Curriculum Learning (DCL) to adaptively adjust the sampling strategy and loss weight in each batch, which achieves the better ability of generalization and discrimination.

In our work, we propose three different strategies to determine the difficulty of the augmented data and add them to the original data for training in an easy-to-difficult order. As a result, the model can preferentially learn from high-confidence data which contains more accurate information, and thus obtain better performance.

3 Proposed Method

3.1 Overview

Figure 2 shows the overview of our approach, which effectively deals with insufficient data for low-resource NER. The framework consists of two parts: data augmentation and denoising. Data augmentation mainly uses BERT (Devlin et al., 2019) to predict the masked words according to the context and synthesizes new sentences by replacing the words in the original masked position to augment the training set. Furthermore, we denoise the augmented data through curriculum learning (Bengio et al., 2009) to obtain higher quality data. We will introduce the details of each part.

3.2 Data Augmentation via Pre-Trained BERT

In this section, we propose a data augmentation method based on pre-trained BERT for low-resource conditions, which predict words based on the context to generate new synthetic sentences. We apply two

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1131-1142, Hohhot, China, August 13 - 15, 2021.

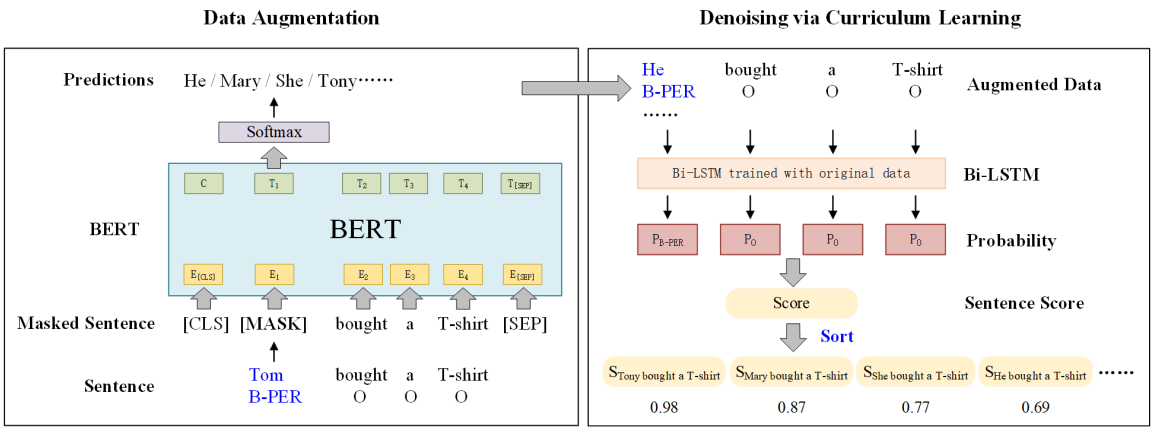

Figure 2: The overall framework. The left is the data augmentation part using pre-trained BERT, and the right is the denoising part using curriculum learning.

methods: *basic method*, which directly uses the BERT model for prediction, and *label-aware method*, which attaches label or label description.

3.2.1 The Basic DA Method

Let a labeled sentence be $\mathbf{x} = (x_1, \cdots, x_m)$ and the corresponding labels be $\mathbf{y} = (y_1, \cdots, y_m)$, where x_i denotes the i-th token and y_i denotes the i-th label. If x_i is inside an entity, which can be judged by its gold label (e.g. B-PER, I-PER, ...), we first mask x_i in this sentence, i.e. masked token $\{x_i\}$ and its context S. Then we use the masked sentence as the input of BERT. The final hidden representation corresponding to the masked token is fed into a softmax layer to generate a sequence of vocabulary size with a probability distribution $p(\cdot|S\{x_i\})$. Then we replace x_i with the k words which have the highest probability. For each sentence in the corpus, we perform the above procedure. Especially note that we only mask the words inside the entity, not the non-entity temporarily. After substituting the masked words with predicted words, our method generates some new sentences, which share the same label sequence with the original sentence. Then these sentences are added to the original low-resource dataset.

3.2.2 Label-Aware Data Augmentation

Although, pre-trained BERT encodes the context information, there is still a lot of noises, such as pronouns and wrong words, in the synthetic sentences. Considering that, we propose a label-aware data augmentation method. Different from basic DA method, this method fine-tunes the BERT before prediction to incorporate label information in the prediction process and improve the matching degree between predicted words and their corresponding labels. We elaborate this process in two steps: *BERT fine-tuning* and *Synthetic Sentence Generation*.

BERT Fine-tuning At first step, we fine-tune the BERT with label-aware original data, which allows us to further train the hidden feature representation with label information. Here we consider two strategies for label-aware data generation:

- Label Additional (LA), where we define all entity types (e.g. PER, ORG, ...) as the training signal.
- Label Description Additional (LDA), where we use descriptions[0] of entity types as the training signal.

As shown in Figure 1, the label (or label description) of masked entity words with [SEP] token is inserted between [CLS] and the first word of the sentence. Then we fine-tune BERT with data obtained from above steps.

[0]We obtain the label descriptions from https://spacy.io/api/annotation#named-entities.

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1131-1142, Hohhot, China, August 13 - 15, 2021.

Algorithm 1 Label-Aware DA Method

Input: Labeled dataset D

Output: Augmented dataset D'

```
for each sentence do
    for each word in sentence do
        if this word in an entity then
            Mask this word.
            Put the label or label description in front of the sentence and separate with [SEP].
            Use BERT to predict words on masked position.
            Replace the original words with the predicted top k words with the highest probability to
            generate new sentences.
        end if
    end for
end for
Add new sentences to the original dataset D to generate an augmented dataset D'.
Perform NER task on augmented dataset D'.
```

Synthetic Sentence Generation At the second step, we use fine-tuned model to make predictions on original data. Note that input data is also transformed into label-aware format, which is the same as fine-tuning data shown in Figure 1 (LA and LDA). Given a masked word x_i and its label y_i, we define its label description as d_i. Different from basic DA method in Section 3.2.1 calculating $p(\cdot|S\{x_i\})$, we calculate $p(\cdot|y_i, S\{x_i\})$ or $p(\cdot|d_i, S\{x_i\})$ in label-aware DA method. The concrete process of label-aware method is shown in Algorithm 1.

3.3 Denoising via Curriculum Learning

BERT is a powerful pre-trained language model, which make full use of contextualized information to generate context-sensitive words. However, directly using original label sequence may cause mismatch between predicted words and original labels, so it is necessary to denoise augmented data via curriculum learning.

3.3.1 Synthetic Example Ranking

As the right part of Figure 2 shows, we train a Bi-LSTM model on original data. Then we use it to predict augmented data without original data and take the predicted probability of the gold label corresponding to each word, i.e. P_i, where i represents the i-th word in a sentence, as the basis for scoring. Based on this process, we artificially formulate three curriculum indicators described in detail as follows:

- **Average.** We calculate the sentence score S_{sent} by averaging the predicted probabilities P_i of all words in it. The lower the value, the more mismatch between the whole sentence and the original gold label, and the more incorrect information contained, which may hurt the model training.

- **Entity.** Different from *average*, we only consider entity words and average their predicted probabilities P_i as sentence score S_{sent}. Entities are more important than other words, because the named entity recognition task is mainly to recognize entities in sentences. Same as above, the higher the value, the more the predicted word matches the original label.

- **Length.** Using sentence length L as the score S_{sent}, we believe that the longer the sentence, the more information it contains, which is more instructive for the training of the model.

Then we sort the sentences in descending order of sentence scores S_{sent}, corresponding to the easy to difficult in the curriculum learning. We believe that prioritizing model to learn more correct information can lead to an improvement in model performance.

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1131-1142, Hohhot, China, Augest 13 - 15, 2021.

3.3.2 Incremental Training

We sample the sorted data according to the ratio of $0\%, 5\%, 10\%, \cdots, 100\%$, and add them to the original data gradually. In the training process, we save the best model on each scale, and the next scale of data uses the model parameters of the previous scale to train. That is to say, we only use a part of synthetic examples to train our model, i.e., samples with high confidence, to reduce the impact of noises in the data augmentation process.

4 Experimental Setups

4.1 Datasets

We consider both the simulated and real-world low-resource scenarios. In the simulated low-resource scenarios, we conduct our experiments on two English NER datasets of different granularities: CoNLL-2003 (Tjong Kim Sang and De Meulder, 2003) and OntoNotes5.0 (Pradhan et al., 2013). CoNLL-2003 is composed of newswire from the Reuters RCV1 corpus and contains four types of named entities: location, organization, person, and miscellaneous. OntoNotes5.0 contains 18 entity types for the CoNLL-2012 task, which includes broadcast conversations, broadcast news, newswire, magazines, telephone conversations, and web texts. Pradhan et al. (2013) comply a core portion of the OntoNotes5.0 dataset and describe a standard train/dev/test split, which we use for our evaluation. As our work mainly focuses on low-resource NER, we randomly select four ratios for CoNLL-2003 and OntoNotes5.0 to simulate the low-resource situation, as shown in Table 1.

CoNLL-2003	**0.2%**	**1%**	**2.5%**	**5%**
Sentence Num	29	149	374	749
OntoNotes5.0	**0.05%**	**0.1%**	**1%**	**2%**
Sentence Num	57	115	1158	2316

Table 1: The sampling ratio of the two datasets and the corresponding sentence number.

In the real-world low-resource scenarios, we conduct our experiments on a dataset from material science: MaScip (Mysore et al., 2019)[1]. This dataset contains 230 synthesis procedures annotated by domain experts with labeled graphs that express the semantics of the synthesis sentences, and 21 entity types (e.g., Material, Number, Operation, Amount-Unit, etc.). We use the train/dev/test split provided by the authors, which contains 1901/109/158 sentences respectively. To simulate a low-resource setting, we also randomly select 50, 150, 500 sentences that contain all entity types from the training set to create the corresponding small, medium, and large training sets (denoted as S, M, L, whereas the complete training set is denoted as F) for each dataset. Note that we only apply data augmentation to the training set, and the development set and test set remain unchanged.

4.2 Implementations

We regard the NER task as a sequence labeling task: given a token sequence, the model needs to predict the label corresponding to each token, which includes position indicators (BIO schema) and entity types. In our study, we use the Bi-LSTM-CRF model (Ma and Hovy, 2016) commonly used in NER tasks as the experimental model. It consists of two parts: a neural-based encoder that creates context-sensitive embedding for each token, whose weights are learned from scratch, the other is a condition random field output layer, which captures the dependencies between adjacent labels. Besides, CNN is used to obtain the character representation of each token, which is then concatenated with the word representation and sent to the bidirectional LSTM layer. The hidden states of the forward and backward LSTM are concatenated together as the final representation. We use a single-layer BiLSTM with a hidden state size of 200. Dropout layers are applied before and after the BiLSTM layer with a dropout rate of 0.5. This model is trained using SGD (Bottou, 2012) with an initial learning rate of 0.015 and batch size of 10. The

[1] https://github.com/olivettigroup/annotated-materials-syntheses

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1131-1142, Hohhot, China, August 13 - 15, 2021.

Method	CoNLL-2003				△	OntoNotes 5.0				△
	0.2%	**1%**	**2.5%**	**5%**		**0.05%**	**0.1%**	**1%**	**2%**	
None	25.72	38.74	50.65	60.65		2.64	12.43	46.71	56.00	
EDA	11.14	33.67	41.86	51.69	-9.3500	7.23	13.82	40.46	48.60	-1.9175
DA	27.11	40.11	**53.23**	59.25	0.9850	12.37	21.43	47.63	**55.39**	4.7600
LA	**29.03**	**42.20**	51.88	**61.48**	2.2075	11.91	**21.46**	**49.29**	54.25	4.7825
LDA	27.90	41.38	52.46	59.91	1.4725	**13.39**	20.19	47.84	**55.39**	4.7575

Table 2: Evaluation results in F_1 score. △ column shows the averaged improvement due to data augmentation. **Bold** means the result is significantly better than the baseline model without data augmentation.

learning rate of each epoch decays proportionally. We use randomly initialized word embeddings with a dimension of 100. We stop training when the F_1 score of the development set has not been updated for 10 epochs. We use the F_1 score to evaluate the effectiveness of the models. The best model saved on the development set is measured using the F_1 score, and finally evaluated on the test set.

4.3 Experimental Results

4.3.1 Impact of Data Augmentation

We compare our methods (DA, LA, LDA) with the following models: None, the original Bi-LSTM-CRF model without data augmentation; EDA (Wei and Zou, 2019), which includes the substitution of synonyms, random insertion, random exchange, and random deletion of words. DA, LA, LDA correspond to the basic method, label additional method, and label description additional method. For each augmentation method, we take $k = 20$ predicted words to replace the masked words.

Table 2 provides the evaluation results on the test set. We can first conclude that our augmentation framework improves over the baseline where no augmentation is used in most cases, and superior to EDA in any ratio. For the CoNLL-2003 dataset, the four proportions increased by 3.31%, 3.46%, 2.58%, and 0.83% respectively compared to the baseline. For the OntoNotes5.0 dataset, the first three proportions have increased by 10.75%, 9.03%, and 2.58% respectively, while the performance of the last proportion has decreased slightly. This situation may reflect the trade-off between the diversity and validity of augmented instances. On the one hand, we use BERT to generate different training instances to prevent overfitting. This positive effect is especially useful when the training set is small. On the other hand, it may also increase the risk of generating invalid instances. For larger training sets, this negative effect may be dominant.

Second, the label additional method outperforms other augmentation on average, i.e. 2.21% for CoNLL-2003 and 4.78% for OntoNotes5.0, although there is no single clear winner across both datasets. However, there is little difference in the performance of the three methods on OntoNotes5.0, which may be due to its more fine-grained entity types. For the label description additional method, the performance is slightly lower than the label additional method. We consider that the label description contains more information compared with the label, which leads to more fixed words predicted by the model and causes a slightly negative impact.

Third, data augmentation techniques are more effective when the training sets are small. In both datasets, data augmentation methods achieve more significant improvements when the training sets are small, such as 0.2% of CoNLL-2003 and 0.05% OntoNotes5.0. In contrast, when the larger training sets are used, the augmentation methods achieve less improvements and some even decrease the performance. This has also been observed in previous work on machine translation tasks (Fadaee et al., 2017).

4.3.2 Impact of Curriculum Learning

Figure 3 shows the results of three different indicators via curriculum learning on CoNLL-2003 and OntoNotes5.0 respectively. We take 1% of CoNLL-2003 LDA data and 0.1% of OntoNotes5.0 LDA

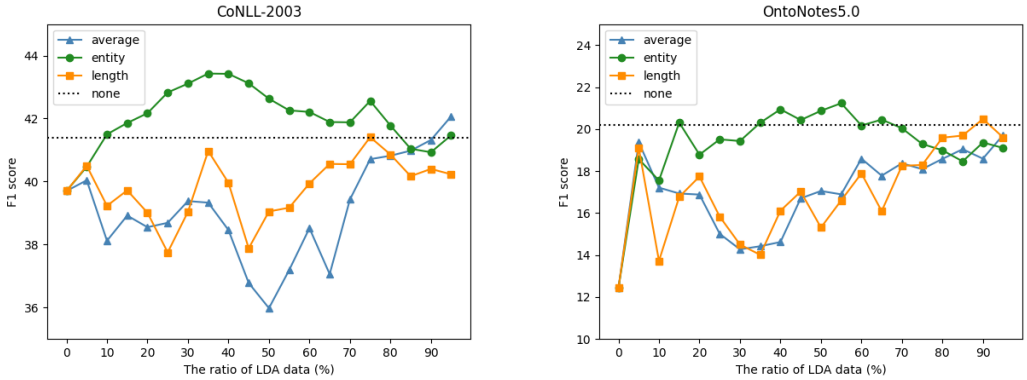

Figure 3: The result of *average*, *entity* and *length* strategies. The black dashed line represents the original LDA method without curriculum learning.

data as examples and use the method described in Section 3.3 for data denoising[2].

As can be seen from the figure, the *entity* indicator is more effective on both datasets. On CoNLL-2003, the best result of 43.43% is achieved on 35% of the LDA augmented data, which is 2.05% higher than the original LDA method. On OntoNotes5.0, the best result of 21.23% is achieved on 55% of the LDA augmented data, which outperforms the original LDA method by 1.04%. It can be explained that when we use data augmentation technologies mentioned in the Section 3.2.1 and 3.2.2, we also introduce noise in the augmented dataset at the same time. The purpose of denoising method is to reduce the negative impact of noise. Experiment results demonstrate the effectiveness of using curriculum learning for data denoising. By giving priority to training high-confidence data except the noise part, the model gets better performance.

4.3.3 Real Low-Resource Scenarios

We use the methods on MaScip, which include data augmentation method via BERT model (DA), label additional data augmentation method (LA), and denoising LA data via curriculum learning (LA-CL), but except the method of additional label description, because this dataset does not have an official description of all labels. Note that when denoising, we only use the *entity* indicator among the above three indicators as it has a best performance.

Method	S	M	L	F
None	59.95	70.05	72.59	76.30
DA	62.08	69.36	72.62	75.03
LA	**62.36**	68.65	73.35	76.31
LA-CL	62.02	**70.41**	**74.31**	**77.29**

Table 3: The result of MaScip[3]. None represent the baseline with unaugmented data.

Table 3 presents comparisons of our methods with the baseline. What we see is that our methods are significantly better than the baseline on the real-world low-resource scenarios. Our methods outperforms the baseline with 62.36%, 70.41%, 74.31% and 77.29% in terms of F1 score. It can be seen that the denoising results have been improved to a certain extent in most of the experiments, including the full amount of data. However, in the small-scale experiment, the denoising result is lower than the LA result. We consider the reason is that the amount of data used to train the Bi-LSTM model for scoring is too small, which makes the model learning less information. This situation leads us to fail to select the most correct predicted words, which negatively feeds back on performance. Therefore, when the training data

[2]For OntoNotes5.0, we do not save the previous scale model, and all start training from scratch.

[3]We leverage GloVe embedding for these experiments.

is particularly small, it is enough to use the data augmentation methods without denoising, by which can achieve an obvious improvement. In summary, the experimental results on the MaScip dataset confirm the effectiveness of our methods in real world low-resource situations.

4.4 Discussion

4.4.1 Performance on Entity Types

To further understand the effectiveness of our method, we investigate the performances for each entity type. Table 4 and Table 5 describe the performance on F1 score for each entity type in 1% of CoNLL-2003 data and 0.1% of OntoNotes5.0 data.

Label	Num	None	DA	LA	LDA
LOC	72	46.04	47.71	**48.98**	47.43
MISC	38	11.04	17.37	**21.78**	20.33
ORG	131	38.93	39.10	**43.55**	40.64
PER	127	39.78	44.50	**45.39**	44.99

Table 4: The F_1 score of each label of CoNLL-2003 (1%).

In CoNLL-2003, it is clear that our methods both outperform the baseline where no augmentation is used on every entity types. Comparing these three methods, we can see that the label additional method is significantly better than the other two. These results first reflect the effectiveness of the label-aware method mentioned in Section 3.2.2. Second, regarding the reason why the LA's score is higher than the LDA's, we consider that the predicted words by LA method are more diverse, which can prevent the model from overfitting.

Label	Num	None	DA	LA	LDA
CARDINAL	16	25.50	**29.79**	27.19	24.68
DATE	41	15.75	32.72	**33.06**	28.13
EVENT	2	0.00	0.00	0.00	**2.11**
FAC	9	0.00	**1.14**	0.00	0.00
GPE	28	13.35	25.05	23.17	**25.31**
LANGUAGE	2	16.67	0.00	8.70	**30.30**
LOC	7	2.52	**10.77**	8.70	9.16
MONEY	12	25.42	**29.76**	28.65	28.49
NORP	4	0.00	7.44	**8.85**	6.05
ORDINAL	2	24.54	**50.24**	30.87	40.25
ORG	82	6.81	10.15	**11.16**	9.51
PERCENT	18	35.12	43.21	56.83	**56.85**
PERSON	37	1.28	8.83	9.69	**12.21**
QUANTITY	2	0.00	0.00	0.00	**0.82**
TIME	6	0.81	1.55	**2.01**	0.76

Table 5: The F_1 score of each label of OntoNotes5.0 (0.1%).

In OntoNotes5.0, due to the complexity of entity types, the three methods have different manifestations in each entity type. It can be seen that the score of some entity types are zero, because the number of sampled sentences is small, and some entity types have almost never appeared. Among them, we omitted some entity types with all zero F_1 score. Summarized as follows, the basic method is better on 'CARDINAL', 'FAC', etc., the label additional method is more effective on 'DETA', 'NORP', 'ORG', and the label description additional method improves more significantly on entity types with lower scores,

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1131-1142, Hohhot, China, Augest 13 - 15, 2021.

Example	Method	Generated Word
[MASK] buys a gaming company.	DA	He She It he **Michae** . **Joe Jack David Tony**
	LA(PER)	He **Sinclair Warner Simon Blackburn Morgan Hamilton Fox Anderson Harry**
	LA(ORG)	**Sony Blackburn** Sinclair He **Dell** Britain **Hamilton Leeds** Fischer **Intel**
[MASK] went bankrupt.	DA	It He They and it he . She which they
	LA(PER)	**Blackburn** Leeds Middlesbrough Barrow Yorkshire York **Hamilton** Italy **Sheffield Milan**
	LA(ORG)	**Salzburg** Switzerland **Zürich** Austria **Blackburn Bavaria Zurich Leeds** Italy **Juventus**
[MASK] plans this event.	DA	Who He Currently She It **FIFA** who **India Nike Israel**
	LA(LOC)	**Canada Ireland Australia** WHO **Bermuda** FIFA UEFA **Argentina Scotland** Yorkshire
	LA(ORG)	**WHO UCI UEFA** Slovenia Slovakia Canada **Yorkshire** Britain Azerbaijan Schedule

Table 6: The table shows the change of the predicted word after using additional label information, where DA means basic method, LA means label additional method, and the brackets indicate the label placed before the sentence. **Bold font** indicates words that match the label. The words from left to right indicate that the predicted probability is from high to low.

like 'EVENT', 'PERSON', 'QUANTITY'. This may be due to the words predicted by the LDA method more closely match the gold label for entity types that appear less frequently.

4.4.2 Case Study

We examine the effect of different methods on generating words. We use 2% of CoNLL-2003 to fine-tune BERT and Table 6 lists some examples of generated words by the basic method and LA method with different settings. All the examples can fill in at least two different entity types of words in the masked position.

As known to all, BERT encodes semantic features from the input sentences for extracting global contexts. In the fist example, when directly using BERT for prediction, we can see that there are some person names generated, which prove the ability of BERT to obtain contextual information. However, one problem with BERT is that the predicted words have a higher probability to be third-person pronouns or even wrong words, which cannot increase the diversity of the augmented data and may hurt the performance. Therefore, we propose the label-aware method described in Section 3.2.2 to minimize this weakness. When we use label additional method with 'PER', more person names appeared, whose probability of being predicted increased. Then when we change the label to 'ORG', some organization names are predicted by the pre-trained model. The situation goes to show the effectiveness of our methods. The same conclusion can be drawn from the second and third examples. In summary, the LA method considers more label information than the DA method, and makes the generated words contain less impurities.

5 Conclusion

In this paper, we propose a novel framework to generate high-quality synthetic data for low-resource NER. We use pre-trained BERT to fully integrate contextual information to generate diverse synthetic sentences, and leverage curriculum learning to denoise synthetic sentences for obtaining higher quality augmented data. Our framework shows superior performance in both simulated low-resource scenarios and real-world low-resource scenarios. In the future, we will explore the performance of our framework when customizing label descriptions and on other token-level NLP tasks.

Acknowledgements

The research work descried in this paper has been supported by the National Key R&D Program of China (2020AAA0108001) and the National Nature Science Foundation of China (No. 61976015, 61976016, 61876198 and 61370130). The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve this paper.

References

Yoshua Bengio, Jérme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *ICML*.

Léon Bottou. 2012. Stochastic gradient descent tricks. In Grégoire Montavon, Genevieve B. Orr, and Klaus-Robert Müller, editors, *Neural Networks*.

Xiang Dai and Heike Adel. 2020. An analysis of simple data augmentation for named entity recognition. In *COLING*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

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 *EMNLP*.

Marzieh Fadaee, Arianna Bisazza, and Christof Monz. 2017. Data augmentation for low-resource neural machine translation. In *ACL*.

Annemarie Friedrich, Heike Adel, Federico Tomazic, Johannes Hingerl, Renou Benteau, Anika Marusczyk, and Lukas Lange. 2020. The SOFC-exp corpus and neural approaches to information extraction in the materials science domain. In *ACL*.

Fei Gao, Jinhua Zhu, Lijun Wu, Yingce Xia, Tao Qin, Xueqi Cheng, Wengang Zhou, and Tie-Yan Liu. 2019. Soft contextual data augmentation for neural machine translation. In *ACL*.

C. Gong, D. Tao, S. J. Maybank, W. Liu, G. Kang, and J. Yang. 2016. Multi-modal curriculum learning for semi-supervised image classification. *IEEE Transactions on Image Processing*, pages 3249–3260.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *ACL*.

Xiaochuang Han and Jacob Eisenstein. 2019. Unsupervised domain adaptation of contextualized embeddings for sequence labeling. In *EMNLP*.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *CoRR*.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *NAACL*.

Canasai Kruengkrai, Thien Hai Nguyen, Sharifah Mahani Aljunied, and Lidong Bing. 2020. Improving low-resource named entity recognition using joint sentence and token labeling. In *ACL*.

Onur Kuru, Ozan Arkan Can, and Deniz Yuret. 2016. Charner: Character-level named entity recognition. In *COLING*.

Cao Liu, Shizhu He, Kang Liu, and Jun Zhao. 2018. Curriculum learning for natural answer generation. In *IJCAI*.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *ACL*, August.

Joel Mathew, Shobeir Fakhraei, and José Luis Ambite. 2019. Biomedical named entity recognition via reference-set augmented bootstrapping. *arXiv preprint arXiv:1906.00282*.

T. Matiisen, A. Oliver, T. Cohen, and J. Schulman. 2020. Teacher–student curriculum learning. *IEEE Transactions on Neural Networks and Learning Systems*, pages 3732–3740.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, pages 39–41.

Junghyun Min, R. Thomas McCoy, Dipanjan Das, Emily Pitler, and Tal Linzen. 2020. Syntactic data augmentation increases robustness to inference heuristics. In *ACL*.

Sheshera Mysore, Zachary Jensen, Edward Kim, Kevin Huang, Haw-Shiuan Chang, Emma Strubell, Jeffrey Flanigan, Andrew McCallum, and Elsa Olivetti. 2019. The materials science procedural text corpus: Annotating materials synthesis procedures with shallow semantic structures. In *Proceedings of the 13th Linguistic Annotation Workshop*.

Minlong Peng, Xiaoyu Xing, Qi Zhang, Jinlan Fu, and Xuanjing Huang. 2019. Distantly supervised named entity recognition using positive-unlabeled learning. In *ACL*.

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1131-1142, Hohhot, China, Augest 13 - 15, 2021.

Anastasia Pentina, Viktoriia Sharmanska, and Christoph H. Lampert. 2015. Curriculum learning of multiple tasks. In *CVPR*.

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom Mitchell. 2019. Competence-based curriculum learning for neural machine translation. In *NAACL*.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using ontonotes. In *CoNLL*.

Jonathan Raiman and John Miller. 2017. Globally normalized reader. In *EMNLP*.

Sebastian Ruder. 2019. *Neural transfer learning for natural language processing*. Ph.D. thesis.

Jingbo Shang, Liyuan Liu, Xiaotao Gu, Xiang Ren, Teng Ren, and Jiawei Han. 2018. Learning named entity tagger using domain-specific dictionary. In *EMNLP*.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *CoNLL*.

Xinyi Wang, Hieu Pham, Zihang Dai, and Graham Neubig. 2018. SwitchOut: an efficient data augmentation algorithm for neural machine translation. In *EMNLP*.

Yiru Wang, Weihao Gan, Jie Yang, Wei Wu, and Junjie Yan. 2019. Dynamic curriculum learning for imbalanced data classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *EMNLP)*.

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. 2020. Unsupervised data augmentation for consistency training. *NIPS*.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension. *CoRR*.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *EMNLP*.

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1131-1142, Hohhot, China, Augest 13 - 15, 2021.