Low-Resource NER by Data Augmentation With Prompting

Jian Liu , **Yufeng Chen** and **Jinan Xu**
Beijing Key Lab of Traffic Data Analysis and Mining
Beijing Jiaotong University, School of Computer and Information Technology, China
jianliu@bjtu.edu.cn, chenyf@bjtu.edu.cn, jaxu@bjtu.edu.cn

Abstract

Named entity recognition (NER) is a fundamental information extraction task that seeks to identify entity mentions of certain types in text. Despite numerous advances, the existing NER methods rely on extensive supervision for model training, which struggle in a low-resource scenario with limited training data. In this paper, we propose a new data augmentation method for low-resource NER, by eliciting knowledge from BERT with prompting strategies. Particularly, we devise a label-conditioned word replacement strategy that can produce more label-consistent examples by capturing the underlying word-label dependencies, and a prompting with question answering method to generate new training data from unlabeled texts. The experimental results have widely confirmed the effectiveness of our approach. Particularly, in a low-resource scenario with only 150 training sentences, our approach outperforms previous methods without data augmentation by over 40% in F1 and prior best data augmentation methods by over 2.0% in F1. Furthermore, our approach also fits with a zero-shot scenario, yielding promising results without using any human-labeled data for the task.

1 Introduction

Named entity recognition (NER), an essential information extraction task, aims to identify named entities of certain types (e.g., PERSON) in texts [Grishman and Sundheim, 1996]. The state-of-the-ar methods for NER are based on supervised learning, demanding a good amount of labeled data for model training [Lample *et al.*, 2016]. However, in a real-world scenario, it is prohibitively expensive to collect large sets of labeled data in many domains (e.g., bio-medicine, military), which significantly limits the applicability of existing NER methods [Mayhew *et al.*, 2017; Hou *et al.*, 2020].

Recently, there is a rise of interest in investigating *data augmentation* (DA) methods to address tasks in low-resource scenarios [Wei and Zou, 2019; Xie *et al.*, 2019]. For natural language processing (NLP), the most successful DA methods are based on word manipulation, showing good performance in sentence-level tasks such as text classification,

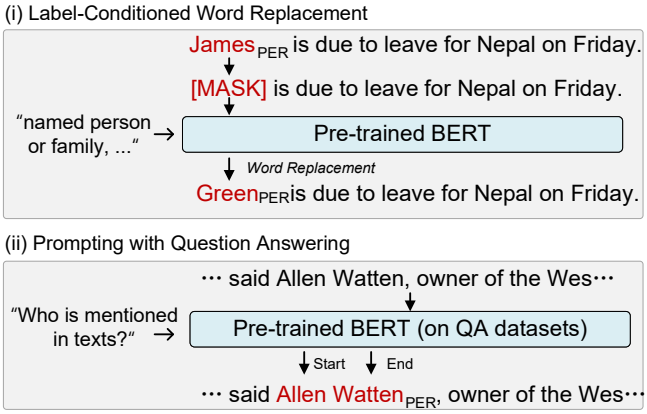

Figure 1: Illustration of our DA method for low-resource NER, by eliciting knowledge from BERT with two prompting mechanisms.

paraphrase identification, and others [Wei and Zou, 2019; Xie *et al.*, 2019; Chen *et al.*, 2020]. However, it is challenging to apply such DA methods to NER, a token-level task sensitive to word manipulation — for example, a word substitution operation may result in a mismatch between the word and the original label [Dai and Adel, 2020; Ding *et al.*, 2020].

In this paper, we introduce a new DA approach for low-resource NER, with knowledge elicitation from BERT [Devlin *et al.*, 2019]. Intuitively, we may easily use BERT to create a new training example by "altering" an existing one. For instance, given a labeled sentence "James is due to leave for Nepal" with "James" marked as a PERSON, we can replace "James" with a placeholder [MASK], and then use BERT to fill in the blank. When BERT predicts a person name "Green", we obtain a perfectly new training example "Green is due to leave for Nepal". However, we may also encounter noisy case because BERT simply evaluates whether a word fits into contexts, such as "He is due to leave for Nepal", where "He" fits in the context well but is not a name.

To address the above issue, we devise two prompting mechanisms for better training data generation, and an uncertainty driven method for noise reduction. Our first prompting mechanism is a label-conditioned word replacement strategy, which incorporates the definition of an entity type (e.g., "name of person, family, ..., and fiction" for PERSON) into the word replacement process; we show that this strategy

can effectively capture label-word dependencies and generate label-consistent words. Our second prompting strategy uses a question answering (QA) style prompt, which can directly generate new training samples from unlabeled texts. For example, we may use a prompt question "Who is referenced in texts?" to query a BERT model pre-trained on QA datasets for identifying PERSON entities in a sentence. We show this strategy is supplementary to the prior one and can yield promising results even in a zero-shot scenario. In addition, we devise an uncertainty-guided self-training method to reduce noise in the generated data. In particular, we set up an iterative framework for training in which at each step only credible examples, as judged by an uncertainty mechanism [Gal and Ghahramani, 2016], are supplied to the training set. We show this method can considerably reduce the effects of noise in the data and robustify the learning process.

The experimental results have well confirmed the effectiveness of our method. Particularly, in a low-resource scenario with only 150 labeled examples, our approach scores 61.2%, 36.2%, and 71.4% in F1 on three benchmarks, outperforming prior best DA methods by more than 2.0% in F1 absolutely and methods without DA by more than 40% in F1. Furthermore, our method yields encouraging results in a zero-shot scenario without the requirement of any human labeled data. We have made our code available at https://github.com/jianliu-ml/fewNER for further investigation.

To summarize, we have three contributions:

- We investigate a new DA method for low-resource NER, based on knowledge elicitation from BERT using two prompting mechanisms. As the first study to investigate the use of prompting mechanisms for DA in NER, our work may inspire more studies in this research line.
- We devise two prompting strategies, based on label-conditioned word replacement and question answering respectively, for training data generation. We also devise an uncertainty-guided self-training strategy for noise reduction. To the best of our knowledge, this is the first effort introducing uncertainty modeling in DA for NER.
- We report promising results on three real-world NER datasets and conduct extensive analysis to provide insight into the reasons behind our method's efficacy.

2 Related Work

Low-Resource NER. The question of how to train a NER model in low-resource environments remains an unresolved one in NLP. Existing research has addressed low-resource NER with distantly supervision [Plank and Agić, 2018], cross-lingual projection [Mayhew *et al.*, 2017; Xie *et al.*, 2018], and meta-learning [Hou *et al.*, 2020], which however require huge domain expertise. Motivated by the success of data augmentation (DA) in computer vision [Wei and Zou, 2019], several works have studied DA for low-resource NER. Particularly, [Dai and Adel, 2020] use label-wise token replacement, synonym replacement, and mention replacement to generate new training examples; [Ding *et al.*, 2020] train a language modeling objective combining words and labels to generate new examples. However, because NER is a fine-grained token-level task, the above approaches often need sophisticated criteria to ensure word-label consistency, which limits their applicability. Unlike previous works, we present a new DA approach for low-resource NER that can elicit knowledge from BERT and does not require considerable human intervention. Our method can generate more reliable instances while simultaneously considering noise reduction through an uncertainty-guided self-training mechanism.

The Prompting Mechanism for Learning. The "pre-train, prompt and predict" paradigm has recently gained popularity in NLP research [Liu *et al.*, 2021], which typically builds a template prompt with some slots and then use a pre-trained language model (e.g., BERT [Devlin *et al.*, 2019]) to fill the slots for performing a task. The existing studies have apply this paradigm to address tasks including open-domain question answering, text classification, and others [Jiang *et al.*, 2020; Schick and Schütze, 2021]. To the best of our knowledge, this is the first study to investigate data augmentation via prompting methods for low-resource NER. Our approach may inspire future work on other related tasks, such as low-resource relation extraction and event extraction.

3 Approach

Figure 2 depicts the overview of our method, which includes two prompting regimes for new training data generation, and an uncertainty-guided self-training strategy for noise reduction. The technical details of our approach follow.

3.1 Label-Conditioned Word Replacement

Let (X, Y) be a labeled example with $X = [w_1, ..., w_n]$ being a sentence of n words and $Y = [l_1, ..., l_n]$ being the entity label sequence in BIO schema. Our goal is to create a new sentence $\hat{X}$ that matches the original label sequence Y by replacing some words in X. Here we propose a label-conditioned pre-training mechanism to capture word-label dependencies — after randomly masking a word in the sentence, we re-train a BERT model to recover it, but use the definition[1] of the entity type as a prompt. Assuming "James is due to leave for Nepal" is a training sentence to re-train BERT and if James is masked, we construct the following input to BERT:

$$\texttt{[CLS]}\ \mathrm{pmt}\ \texttt{[SEP]}\ \overbrace{\texttt{[MASK]}\ \text{is due to leave for Nepal}}^{\text{The masked sentence}} \quad (1)$$

where $\mathrm{pmt}_{\mathrm{l_i}}$ indicates the definition of PERSON, i.e., "Named person or family". In this way, the recovery of a word is conditioned not only on contexts, but also on the entity label, and therefore the BERT mode is tuned to favor words matching the labels. For new data generation, we randomly mask a word in a sentence and generate a substitution by sampling from the predictive probabilities. We limit the number of generated examples to T for each sentence.

3.2 Prompting with Question Answering

Given that label-conditioned word replacement still requires some labeled data for model training and can only generate

[1] We obtain the entity type definitions from the spacy project and use "other" as a prompt for words labeled as O.

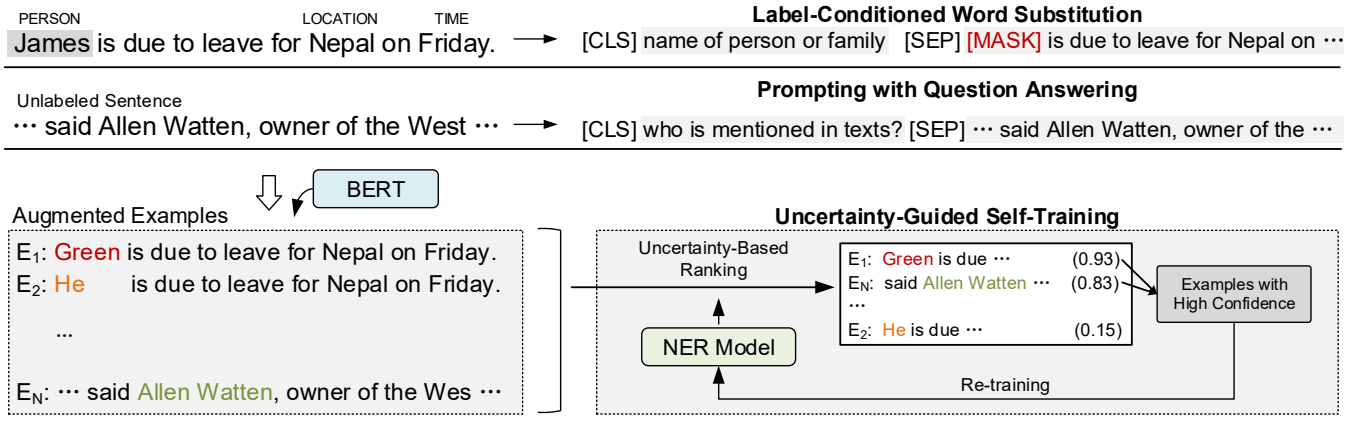

Figure 2: The overview of our approach, which contains two prompting mechanisms for new training data generation (top) and an uncertainty-guided self-training regime for noise reduction (bottom).

Type	Strategy	Prompt Representation
PER	Label Condition	"Named person or family."
	Query Style	"Who is the named person or family mentioned in texts?"
LOC	Label Condition	"Name of politically or geographically defined location."
	Query Style	"Which politically or geographically ... are mentioned in texts?"

Table 1: The prompt representations of different methods.

instances similar to the existing ones, we devise a query-style prompting mechanism that can generate novel examples from unlabeled texts. The central idea is to use inquiry questions as prompts to "annotate" entities in texts, based on a BERT model pre-trained on large scale question answering (QA) datasets [Rajpurkar *et al.*, 2018]. For example, to recognize PERSON entities in plain texts, we create a query: *Who is the named person or family mentioned in texts?* (derived from the definition of PERSON) and use it as a prompt to query a BERT model. Particularly, we create the following sequence:

$$\texttt{[CLS]}\ \text{query question}\ \texttt{[SEP]}\ \text{unlabeled sentence} \quad (2)$$

and encode it to $\boldsymbol{H} \in \mathcal{R}^{N \times d}$ with N being the sequence's length and d being BERT's dimension. Then we locate a PERSON entity by computing two vectors specifying the probabilities of the starting and ending positions:

$$\boldsymbol{p}_{start} = \text{softmax}(\boldsymbol{H}\boldsymbol{w}_{\text{start}} + \boldsymbol{b}_{\text{start}}) \quad (3)$$

$$\boldsymbol{p}_{end} = \text{softmax}(\boldsymbol{H}\boldsymbol{w}_{\text{end}} + \boldsymbol{b}_{\text{end}}) \quad (4)$$

where $\boldsymbol{w}_{\text{start}} \in \mathcal{R}^d$, $\boldsymbol{b}_{\text{start}} \in \mathcal{R}^N$, $\boldsymbol{w}_{\text{end}} \in \mathcal{R}^d$, and $\boldsymbol{b}_{\text{end}} \in \mathcal{R}^N$ are mode parameters. We consider the identified entities, coupled with the texts, as new training data. Table 1 compares the prompt representations of different prompting methods.

3.3 Noise Reduction via An Uncertain-Guided Self-Training Strategy

Despite their effectiveness, the above methods can introduce noise because no hard constraints are imposed. We develop a self-training framework [Scudder, 1965] combining uncertainty modeling [Gal and Ghahramani, 2016] for noise reduction. Let the labeled dataset be $\mathcal{D}_L$, and the automatically generated dataset by the two prompting strategies be $\mathcal{D}_U$. Our self-training framework repeats the following steps:

1. Train a NER model M on $\mathcal{D}_L$.
2. Use M to pseudo-annotate examples in $\mathcal{D}_U$.
3. Select a set of *reliable examples* from $\mathcal{D}_U$ and move them into $\mathcal{D}_L$; repeat the above steps until convergence.

Defining a universally good criterion for selecting reliable examples in step 3) remains an open challenge. Here we devise an uncertainty-guided method [Gal and Ghahramani, 2016] with an exploration-exploration trade-off strategy.

Uncertainty-Guided Confidence Ranking. We leverage recent advances in uncertainty modeling [Gal and Ghahramani, 2016] to evaluate the reliability of an example. Assume $(\hat{X}, \hat{Y}) \in \mathcal{D}_U$ is an automatically generated example, with $\hat{X} = [\hat{w}_1, ..., \hat{w}_n]$ being the sentence and $\hat{Y} = [\hat{l}_1, ..., \hat{l}_n]$ being the entity label sequence. We first pseudo-predict a label for each word using the current NER model, with dropout layers activated and perform K times in total. Assume $L^i = [\tilde{p}^i_1, ..., \tilde{p}^i_K]$ is the obtained set containing K predicted labels[2] for the i^{th} word $\tilde{w}_i$. According to [Gal and Ghahramani, 2016], the variance of L^i reflects the model's uncertainty on its prediction for $\hat{w}_i$. Given this, we define a token-wise criterion measuring the current model's confidence that $\hat{w}_i$ matches $\hat{l}_i$:

$$\text{C}_{\text{token}}(\hat{w}_i, \hat{l}_i) = n(\hat{l}_i)/K \quad (5)$$

where $n(\hat{l}_i)$ is frequency of $\hat{l}_i$ in L^i. Based on this token-wise criterion, we then define a sentence-level criterion to assess the degree of match between $\hat{X}$ and $\hat{Y}$:

$$\text{Certainty}(\hat{X}, \hat{Y}) = \frac{1}{n}\sum_{i=1}^{n} \text{C}_{\text{token}}(\hat{w}_i, \hat{l}_i) \quad (6)$$

where n is the length of $\hat{X}$. A higher score implies the current model is more certain that $\hat{X}$ and $\hat{Y}$ complement each other.

[2]Due to the activated dropout layers, elements in L^i may differ.

Exploration-Exploitation Trade-Off. In practice, always selecting the most certain examples may not be a good solution, because the certain-most examples are too easy to contribute additional information, and this usually slows down learning and results in sub-optimal performance (as shown in our experiments). To balance speed and efficiency, we design an exploration-exploiting trade-off strategy, by defining a selection weight for each example:

$$w_{(\hat{X},\hat{Y})} = \frac{\exp^{\text{Certainty}(\hat{\text{X}},\hat{\text{Y}})}}{\sum_{X,Y\in\mathcal{D}_U} \exp^{\text{Certainty}(\text{X},\text{Y})}} \quad (7)$$

We sample out N examples based on the above weight at each training iteration — this method biases the selection process towards picking more certain samples for exploiting but also allows for picking less certain examples for exploration. In evaluations, we show this method outperforms both certain-first and uncertainty-first strategies.

4 Experimental Setups

Datasets and Evaluations. We use three NER datasets for evaluation: CoNLL 2003 [Tjong Kim Sang and De Meulder, 2003], OntoNotes 5.0 [Hovy *et al.*, 2006], and a real-world low-resource dataset, MaScip [Mysore *et al.*, 2019]. Among them, CoNLL 2003 defines four coarse-grained entity types: PER (Person), LOC (Location), ORG (Organization), and MISC (Miscellaneous); OntoNotes 5.0 defines 18 fine-grained types, including CARDINAL, MONEY, PRODUCT, and others; MaScip [Mysore *et al.*, 2019] defines 21 entity types (e.g., Material, Number, Operation, Amount-Unit) involved in materials synthesis procedures. Table 2 gives statistics of the three datasets. As for evaluation, following [Ding *et al.*, 2020], for each dataset we sample out 50, 150, and 500 sentences (at least one mention of each entity type is included) to create the small (S), medium (M), and large (L) training sets (we use F to indicate the full training set), and we use precision (P), recall (R), and F1 as evaluation metrics.

Implementations. In our approach, we use `BERT-base` cased version [Devlin *et al.*, 2019] as the backbone considering NER is a case-sensitive task. In label-conditioned word replacement, we empirically set T = 10, meaning to expand the original labeled data set by 10 times. In our prompting with question answering method, we train a BERT model on SQuAD datasets [Rajpurkar *et al.*, 2018] and sample out 100,000 unlabeled sentences from WikiPedia for new data generation (For example, on CoNLL 2003, the numbers of retrieved entities for each type are PER:5633, ORG:3978, LOC: 2342, MISC: 1197). In the uncertainty-guided self-training method, we set the number of forward passes K to 10, chosen from [5, 10, 20] and select N = 200 (chosen from [50, 100, 200, 300]) examples at each iteration. We use the development set to tune the best iteration step. We evaluate the impact of our method on two typical NER models: (i) BiLSTM-CRF [Lample *et al.*, 2016], which combines Long Short-Term Memory (LSTM) networks [Hochreiter and Schmidhuber, 1997] with a CRF tagger [Lafferty *et al.*, 2001] for NER. We use GloVe embeddings [Pennington *et al.*, 2014] and we set the batch size to 50 (chosen from [10, 20, 50, 100]) and the learning rate to 1e-2 (chosen from [1e-1, 1e-2, 1e-3, 1e-4]). (ii) BERT based entity tagger (to clear confusion, we denote it as Transformer). The batch size is set to 10 (chosen from [2, 5, 10, 20]), and the learning rate is set to 1e-5 (chosen from [5e-6, 1e-5, 5e-5, 1e-4]). We apply Adam [Kingma and Ba, 2015] for model optimization.

Dataset	Split	# Sen.	# Token	# Entity
CoNLL 2003	Train	14,987	204,567	23,499
	Dev	3,466	51,578	5,942
	Test	3,684	46,666	5,648
OntoNotes 5.0	Train	59,924	1,088,503	81,828
	Dev	8,528	147,724	11,066
	Test	8,262	152,728	11,257
MaScip	Train	1,910	61,750	18,874
	Dev	109	4,158	1,190
	Test	158	4,585	1,259

Table 2: Data statistics of the NER datasets used in this study.

DA Methods for Comparison. We compare our approach with the following DA methods: 1) EDA [Wei and Zou, 2019], a DA method for text classification, which adopts synonymous word substitution, word insertion, and shuffling for data generation. [Ding *et al.*, 2020] adapt it to low-resource NER. 2) BDA [Zhou *et al.*, 2019], a DA method for text classification that directly uses BERT for lexical substitution, which does not consider label information. 3) CBT [Kumar *et al.*, 2020], a DA method for text classification that leverages sentence-level labels as condition signals. We adapt this method for NER using entity labels (rather than label definitions) as condition signals. 4) FlipDA [Zhou *et al.*, 2021], a DA method using a generative model and a classifier to generate label-flipped data. We denote our approach by PromptDA and use LC and QA to indicate label-conditioned word replacement and prompting with QA respectively. We use UC to indicate the uncertainty-guided self-training method.

5 Experimental Results

Table 3 summarizes the results on CoNLL 2003, OntoNotes 5.0, and MaSciP, using the small (S), medium (M), large (L), and full (F) settings. The effectiveness of our approach has been justified. For example, in the M setup with 150 training examples only, our approach (with BiLSTM-CRF architecture) scores 61.2%, 36.2%, and 71.4% in F1 on the three datasets, outperforming previous methods without DA by more than 40% in F1 and prior best DA methods by 2.0% in F1. Furthermore, we note that the two prompting strategies are complementary, and that their combination produces the best result. When we compare the two prompting strategies, we note label conditioned word replacement (LC) is more effective than prompting with question answering (QA), which could be due to the latter strategy introducing more noise.

Impact of Uncertainty-Guided Self-Training. Figure 3 compares our uncertainty-guided strategy (denoted by U) with easy-first (E) strategy, hard-first (H) strategy, and a method training on all generated data (ALL). The performance is based on the development sets of CoNLL 2003

Method	CoNLL 2003				OntoNotes 5.0				MaSciP			
	S	M	L	F	S	M	L	F	S	M	L	F
BiLSTM+CRF	-	13.3	27.2	**91.7**	-	0.2	17.1	88.0	52.4	70.0	72.6	76.3
w/ EDA [Ding *et al.*, 2020]	6.3	45.5	62.4	90.1	8.3	11.7	42.9	85.2	58.4	66.3	72.1	75.8
w/ BDA [Zhou *et al.*, 2019]	13.4	58.8	64.1	90.6	12.8	33.9	49.5	87.7	62.1	69.4	72.6	76.1
w/ CBT [Kumar *et al.*, 2020]	13.1	59.2	63.7	91.3	13.5	34.8	49.2	87.5	62.4	68.7	73.4	76.5
w/ PromptDA (LC)	12.6	51.4	55.8	91.0	13.3	31.6	48.6	87.1	60.1	69.2	73.1	77.0
w/ PromptDA (QA)	11.3	41.0	53.1	90.8	12.2	29.8	41.2	87.5	58.4	68.7	73.3	77.0
w/ PromptDA (LC+QA)	14.9	60.0	66.7	91.5	13.7	35.5	50.4	**88.2**	**62.7**	70.2	73.3	76.9
w/ PromptDA (LC+QA) + UC	**15.2**	**61.2**	**67.2**	91.2	**14.1**	**36.2**	**51.1**	87.5	62.0	**71.4**	**74.3**	**77.3**
Transformer	-	55.0	58.8	92.4	-	3.7	52.4	**90.3**	57.6	71.1	73.0	76.5
w/ EDA [Ding *et al.*, 2020]	1.1	61.4	64.2	91.3	5.9	22.1	54.8	87.0	58.8	70.0	72.9	76.0
w/ BDA [Zhou *et al.*, 2019]	35.6	65.9	68.4	92.0	11.6	39.2	56.7	89.1	59.9	71.5	73.3	76.8
w/ CBT [Kumar *et al.*, 2020]	35.5	66.0	68.9	91.5	12.4	41.9	58.2	89.5	60.1	70.7	72.4	77.0
w/ FlipQA [Zhou *et al.*, 2021]	43.7	66.5	69.6	91.1	12.3	42.0	58.3	89.4	61.0	70.3	71.1	77.2
w/ PromptDA (LC)	33.1	58.2	67.0	91.6	12.1	40.1	57.7	87.2	63.0	71.4	73.1	77.3
w/ PromptDA (QA)	23.5	56.9	66.9	92.0	11.7	38.8	56.1	87.9	61.2	70.9	73.1	77.8
w/ PromptDA (LC+QA)	43.4	66.9	69.7	91.9	12.4	42.4	58.8	87.6	63.2	71.7	**73.4**	76.9
w/ PromptDA (LC+QA) + UC	**44.1**	**67.2**	**70.1**	**92.5**	**13.2**	**42.8**	**59.3**	88.3	**72.7**	**71.9**	73.2	**78.1**

Table 3: Results (a 5-run average) on CoNLL 2003, OntoNotes, and MaScip. The best scores are in **bold** and the second best are underlined.

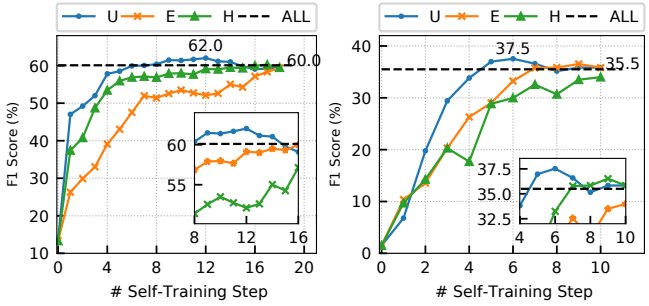

Figure 3: Impact of our uncertainty-guided self-training strategy on CoNLL 2003 (left) and OntoNotes 5.0 (right).

Dataset	Entity Type	w/o DA	CBT	LC	LC+QA
CoNLL	PER	$64.9_{\pm 0.1}$	$73.4_{\pm 0.1}$	$76.0_{\pm 0.2}$	$\mathbf{77.0}_{\pm 0.4}$
	ORG	$41.8_{\pm 1.5}$	$50.9_{\pm 0.2}$	$51.3_{\pm 1.3}$	$\mathbf{52.0}_{\pm 1.1}$
	LOC	$33.6_{\pm 2.4}$	$60.0_{\pm 0.4}$	$59.5_{\pm 0.1}$	$\mathbf{61.1}_{\pm 0.2}$
	MISC	$\mathbf{33.6}_{\pm 1.6}$	$31.0_{\pm 0.2}$	$31.3_{\pm 0.3}$	$31.0_{\pm 1.1}$
OntoNotes	CARDINAL	$11.2_{\pm 2.7}$	$31.2_{\pm 1.1}$	$33.0_{\pm 0.4}$	$\mathbf{33.1}_{\pm 0.9}$
	GPE	$2.5_{\pm 2.2}$	$40.0_{\pm 2.7}$	$42.3_{\pm 3.1}$	$\mathbf{49.4}_{\pm 0.2}$
	MONEY	$10.1_{\pm 1.1}$	$38.2_{\pm 1.6}$	$40.7_{\pm 2.1}$	$\mathbf{45.8}_{\pm 0.9}$
	NOPR	$40.5_{\pm 2.5}$	$39.3_{\pm 1.3}$	$39.3_{\pm 4.2}$	$\mathbf{41.0}_{\pm 0.8}$
	ORDINAL	$20.2_{\pm 3.4}$	$44.0_{\pm 1.6}$	$42.6_{\pm 1.0}$	$\mathbf{46.5}_{\pm 2.4}$
	DATE	$7.7_{\pm 3.0}$	$\mathbf{37.9}_{\pm 0.9}$	$36.1_{\pm 0.6}$	$36.8_{\pm 1.3}$
	PERCENT	$16.6_{\pm 1.1}$	$\mathbf{58.9}_{\pm 0.9}$	$58.1_{\pm 1.6}$	$57.8_{\pm 0.1}$
MaSciP	Material	$66.1_{\pm 1.6}$	$61.2_{\pm 0.5}$	$68.3_{\pm 0.4}$	$\mathbf{69.5}_{\pm 0.9}$
	Brand	$64.9_{\pm 2.2}$	$65.1_{\pm 2.1}$	$67.2_{\pm 3.5}$	$\mathbf{69.7}_{\pm 0.2}$
	Nonrecipe	$57.5_{\pm 3.3}$	$62.3_{\pm 1.2}$	$\mathbf{67.9}_{\pm 1.2}$	$66.3_{\pm 1.3}$
	Number	$66.5_{\pm 1.1}$	$\mathbf{68.1}_{\pm 2.4}$	$65.2_{\pm 4.1}$	$65.8_{\pm 0.9}$
	Amount-Unit	$56.6_{\pm 1.1}$	$\mathbf{58.2}_{\pm 1.1}$	$55.4_{\pm 2.8}$	$55.8_{\pm 0.9}$

Table 4: Performance breakdown on two typical entity types.

and OntoNotes 5.0. From the results, our uncertainty-guided method outperforms other methods in terms of both convergence speed and the final F1. For example, on CoNLL 2003, it achieves the highest F1 with 8 to 14 training steps and outperforms other methods by at most 2.0% in F1. We observe interesting results by comparing the easy-first and hard-first strategies: The easy-first strategy, which always selects the most certain examples for training, results in the slowest learning rate, maybe because the most certain cases are too easy to provide more knowledge for learning. However, the hard-first strategy demonstrates contradictory behaviors: it performs well on CoNLL 2003 but poorly on OntoNotes 5.0. Because the hard-first strategy always chooses the most uncertain data for training, it can quickly capture complementary patterns to the existing data for CoNLL 2003, which has fewer entity types and more clean generated data. On the other hand, it suffers from noise for OntoNotes 5.0, which has more entity types and more noisy generated examples.

6 Discussion

We perform in-depth studies to explore the reasons behind our approach's effectiveness. To simplify discussion, we choose the medium training set (M) setting.

Per-Type Performance. Table 4 shows performance breakdown on different entity types. Particularly, on CoNLL 2003, our method performs well on PER, ORG, and LOC, but MISC. The reason is that MISC's definition is relatively vague (i.e., "events, nationalities, products, or works of art") and does not have a consistent pattern, which degrades our method using entity type definition as prompt. On OntoNotes 5.0, our method yields better performance for common types such as GPE and MONEY, but value types such as DATE and PERCENT; we observe a similar pattern

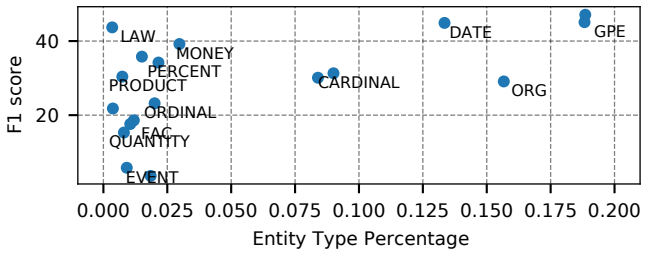

Figure 4: Results of zero-shot NER on OntoNotes 5.0.

on MsScip, with performance on Material and Brand being better than performance on value types such as Number and Amount-Unit. The underlying reason is that common entity types such as GPE and MONEY have more clear and expressive definitions than value types such as DATE ("times smaller than a day") and PERCENT ("percentage, including %"), and such expressive definitions provide more prior knowledge to instruct our method for data augmentation.

Results of Zero-Shot NER. Considering that our prompting with question answering strategy can directly generate new training data from unlabeled texts, we study whether it can be applied to the zero-shot scene. Figure 4 shows the type-wise F1 (each type is evaluated independently) of the BiLSTM+CRF model on OntoNotes 5.0, trained on the automatically generated data solely. Our method demonstrates very promising results, for example, achieving over 40% in F1 on PER, GPE, LAW, and DATE without any labeled data. By comparing different entities types, we show that our method benefits both common types and rare types (such as MONEY, PERCENTAGE, PRODUCT) as long as they have clear definitions, but on types having a vague definition such as EVENT.

Data Fidelity Check. Table 5 explores data fidelity check, by comparing the words generated by BDA, CBT, and our label conditioned prompting approach (LC). By exploring E1 and E2, we show our method generates more label-consistent words (with a ratio of 80% and 90%) than BDA and CBT. E3 and E4 are more interesting cases, where we deliberately design ambiguous contexts and give models different labels for word generation (For example, in E3, the [MASK] can be filled with either a PER or ORG entity). From the results, our approach outperforms other methods for generating words matching the given labels — particularly, in E4, even an incorrect LOC label is given, our method generate wrong words matching the label, justifying its ability to capture the word-label dependencies. Lastly, in Table 6, we show examples generated by our prompting with QA method. Our approach can yield ideal examples such as E1, E4, E5, E6, E7, but also flawed ones such as E2, E9, and E3, where "Shapiro and Clark" and "Colorado and San Francisco" should be separated as two entities and "someone who is disabled" is not an entity. These cases imply further improvement direction.

7 Conclusion

In this work, we propose a new data augmentation method for low-resource NER, by eliciting knowledge from BERT with two prompting mechanisms. Our method can produce more label-consistent data or new examples from unlabeled texts without considerable human intervention. We also devise an uncertainty-guided self-training method for noise reduction. The results on three datasets have justified our method's effectiveness. In the future, we would apply our method to other related extraction tasks such as relation extraction.

E1: The 53-year-old **[Johnson]**$_{\text{PER}}$ was hospitalized ...		
BDA	man, woman, girl, boy, child ...	(0%)
CBT	man, victim, Collins, Cuban, patient ...	(50%)
LC	Collins, King, Miller, Pope, Wilson ...	(80%)
E2: **[Frankfurt]**$_{\text{LOC}}$ was one bright spot in Europe.		
BDA	That, This, Paris, It, London ...	(60%)
CBT	Rome, It, London, This, Paris ...	(70%)
LC	Paris, London, France, Rome, Italy ...	(90%)
E3: **[MASK]** buys a gaming company ZeniMax.		
BDA	He, She, It, he, Micheal, Joe, Jack ...	
CBT (PER)	He, Trump, Clinton, Obama, Putin ...	
LC (PER)	Brandy, Wilson, Jones, Taylor, Martin ...	
CBT (ORG)	Microsoft, Dell, Russian, Intel, Nokia ...	
LC (ORG)	Microsoft, Nokia, Atari, Nintendo, Intel ...	
E4: **[MASK]** eats the cake.		
BDA	He, She, Everyone, Peter, Alex ...	
CBT (PER)	Everyone, He, She, Everybody, Alex ...	
LC (PER)	Whoever, Oscar, Brady, Clinton, Heinz ...	
CBT (LOC)	Who, Everyone, McDonald, Everybody, Clinton ...	
LC (LOC)	Finland, Turkey, Sweden, Japan, Britain ...	

Table 5: Data fidelity check. The underlined words indicates words that **do not match** the given label.

Type	Generated Examples
PER	E1: More than once, [Judge Kennedy Powell], who is ... E2: The exchange between [Shapiro and Clark] over ... E3: they 're having with [someone who is disabled]
ORG	E4: The [Negro League] is a pleasant diversion from ... E5: [Eli Lilly] shares fell 7, to 50 . E6: ... secretary of the [Communist Party Central Committee] says that ...
LOC	E7: [San Francisco] native Harry Glover Hughes, a ... E8: Nothing could be ... from [Japan]'s agenda. E9: [Colorado and San Francisco] lost their third ...

Table 6: Examples generated by our prompting with QA method.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No.62106016). It is also supported by Fundamental Research Funds for the Central Universities (No. 2021RC234), the National Key R&D Program of China (2019YFB1405200), and the Open Projects Program of National Laboratory of Pattern Recognition.

References

[Chen *et al.*, 2020] Hannah Chen, Yangfeng Ji, and David Evans. Finding Friends and flipping frenemies: Automatic paraphrase dataset augmentation using graph theory. In *Findings of EMNLP*, pages 4741–4751, 2020.

[Dai and Adel, 2020] Xiang Dai and Heike Adel. An analysis of simple data augmentation for named entity recognition. In *COLING*, pages 3861–3867, 2020.

[Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186, 2019.

[Ding *et al.*, 2020] Bosheng Ding, Linlin Liu, Lidong Bing, Canasai Kruengkrai, Thien Hai Nguyen, Shafiq Joty, Luo Si, and Chunyan Miao. DAGA: Data augmentation with a generation approach for low-resource tagging tasks. In *EMNLP*, pages 6045–6057, 2020.

[Gal and Ghahramani, 2016] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. PMLR, pages 1050–1059, 2016.

[Grishman and Sundheim, 1996] Ralph Grishman and Beth Sundheim. Message Understanding Conference- 6: A brief history. In *COLING*, pages 1–20, 1996.

[Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, page 1735–1780, 1997.

[Hou *et al.*, 2020] Yutai Hou, Wanxiang Che, Yongkui Lai, Zhihan Zhou, Yijia Liu, Han Liu, and Ting Liu. Few-shot slot tagging with collapsed dependency transfer and label-enhanced task-adaptive projection network. In *ACL*, pages 1381–1393, 2020.

[Hovy *et al.*, 2006] Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. OntoNotes: The 90% solution. In *NAACL*, pages 57–60, 2006.

[Jiang *et al.*, 2020] Zhengbao Jiang, Antonios Anastasopoulos, Jun Araki, Haibo Ding, and Graham Neubig. X-FACTR: Multilingual factual knowledge retrieval from pretrained language models. In *EMNLP*, pages 5943–5959, 2020.

[Kingma and Ba, 2015] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[Kumar *et al.*, 2020] Varun Kumar, Ashutosh Choudhary, Eunahothers Cho, et al. Data augmentation using pre-trained transformer models. In *Life-long Learning for Spoken Language Systems*, pages 18–26, 2020.

[Lafferty *et al.*, 2001] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, page 282–289, 2001.

[Lample *et al.*, 2016] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In *NAACL*, pages 260–270, 2016.

[Liu *et al.*, 2021] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. In *CoRR*, volume abs/2107.13586, 2021.

[Mayhew *et al.*, 2017] Stephen Mayhew, Chen-Tse Tsai, and Dan Roth. Cheap translation for cross-lingual named entity recognition. In *EMNLP*, pages 2536–2545, 2017.

[Mysore *et al.*, 2019] Sheshera Mysore, Zachary Jensen, Edward Kim, Kevin Huang, Haw-Shiuan Chang, Emma Strubell, Jeffrey Flanigan, Andrew McCallum, and Elsa Olivetti. The materials science procedural text corpus: Annotating materials synthesis procedures with shallow semantic structures. In *LA*, pages 56–64, 2019.

[Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014.

[Plank and Agić, 2018] Barbara Plank and Željko Agić. Distant supervision from disparate sources for low-resource part-of-speech tagging. In *EMNLP*, pages 614–620, 2018.

[Rajpurkar *et al.*, 2018] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In *ACL*, pages 784–789, 2018.

[Schick and Schütze, 2021] Timo Schick and Hinrich Schütze. Exploiting cloze-questions for few-shot text classification and natural language inference. In *EACL*, pages 255–269, 2021.

[Scudder, 1965] H. Scudder. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, pages 363–371, 1965.

[Tjong Kim Sang and De Meulder, 2003] Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *NAACL*, pages 142–147, 2003.

[Wei and Zou, 2019] Jason Wei and Kai Zou. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *EMNLP*, pages 6382–6388, 2019.

[Xie *et al.*, 2018] Jiateng Xie, Zhilin Yang, Graham Neubig, Noah A. Smith, and Jaime Carbonell. Neural cross-lingual named entity recognition with minimal resources. In *EMNLP*, pages 369–379, 2018.

[Xie *et al.*, 2019] Qizhe Xie, Eduard Hovy, Minh-Thang Luong, and Quoc V. Le. Self-training with noisy student improves imagenet classification. In *CVPR*, 2019.

[Zhou *et al.*, 2019] Wangchunshu Zhou, Tao Ge, Ke Xu, Furu Wei, and Ming Zhou. BERT-based lexical substitution. In *ACL*, pages 3368–3373, 2019.

[Zhou *et al.*, 2021] Jing Zhou, Yanan Zheng, Jie Tang, Jian Li, and Zhilin Yang. Flipda: Effective and robust data augmentation for few-shot learning. abs/2108.06332, 2021.