Entity-to-Text based Data Augmentation for various Named Entity Recognition Tasks

Xuming Hu[1*], **Yong Jiang**[2], **Aiwei Liu**[1], **Zhongqiang Huang**[2], **Pengjun Xie**[2], **Fei Huang**[2], **Lijie Wen**[1], **Philip S. Yu**[1,3]
[1]Tsinghua University, [2]Alibaba DAMO Academy, [3]University of Illinois at Chicago
{hxm19,liuaw20}@mails.tsinghua.edu.cn
{yongjiang.jy,chengchen.xpj}@alibaba-inc.com,
wenlj@tsinghua.edu.cn,psyu@uic.edu

Abstract

Data augmentation techniques have been used to alleviate the problem of scarce labeled data in various NER tasks (flat, nested, and discontinuous NER tasks). Existing augmentation techniques either manipulate the words in the original text that break the semantic coherence of the text, or exploit generative models that ignore preserving entities in the original text, which impedes the use of augmentation techniques on nested and discontinuous NER tasks. In this work, we propose a novel *Entity-to-Text* based data augmentation technique named EnTDA to add, delete, replace or swap entities in the entity list of the original texts, and adopt these augmented entity lists to generate semantically coherent and entity preserving texts for various NER tasks. Furthermore, we introduce a diversity beam search to increase the diversity during the text generation process. Experiments on thirteen NER datasets across three tasks (flat, nested, and discontinuous NER tasks) and two settings (full data and low resource settings) show that EnTDA could bring more performance improvements compared to the baseline augmentation techniques.

1 Introduction

Recent neural networks show decent performance when a large amount of training data is available. However, these manually labeled data are labor-intensive to obtain. Data augmentation techniques (Shorten and Khoshgoftaar, 2019) expand the training set by generating synthetic data to improve the generalization and scalability of deep neural networks, and are widely used in NLP (Feng et al., 2021; Li et al., 2022a). One successful attempt for data augmentation in NLP is manipulating a few words in the original text, such as word swapping (Şahin and Steedman, 2018; Min et al., 2020) and random deletion (Kobayashi, 2018; Wei and Zou, 2019). These methods generate synthetic texts effortlessly without considering the semantic coherence of sentences. More importantly, these augmentation approaches work on sentence-level tasks like classification but cannot be easily applied to fine-grained and fragile token-level tasks like Named Entity Recognition (NER).

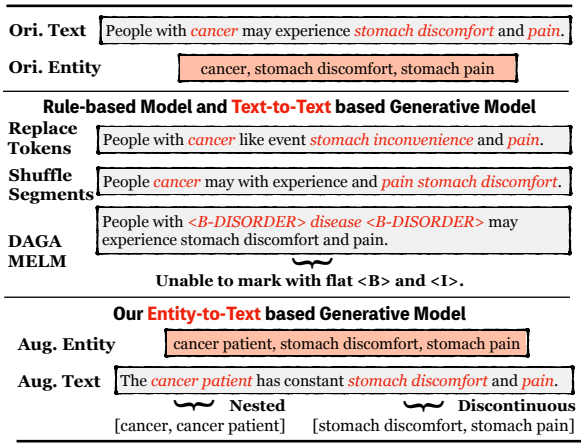


Figure 1: Comparison of augmented cases generated by Rule-based model and *Text-to-Text* based generative model vs. Our *Entity-to-Text* based generative model.

Named Entity Recognition aims at inferring a label for each token to indicate whether it belongs to an entity and classifies entities into predefined types. Due to transformations of tokens that may change their labels, Dai and Adel (2020) augment the token-level text by randomly replacing a token with another token of the same type. However, it still inevitably introduces incoherent replacement and results in syntax-incorrect texts. DAGA (Ding et al., 2020) and MELM (Zhou et al., 2022) investigate the Text-to-Text data augmentation technique using generative methods that preserve semantic coherence and recognize entities through entity tagging during text generation. However, since it is difficult to use flat $\langle B-Type\rangle$ and $\langle I-Type\rangle$ labels to mark nested and discontinuous entities during text generation, these methods can only be used for flat NER tasks. In addition, only the en-

*Work done during an internship at Alibaba DAMO Academy.

Findings of the Association for Computational Linguistics: ACL 2023, pages 9072–9087

tities are masked during the generation process, so that the diversity of generated texts is also limited. For example, as shown in Figure 1, rule-based models replace tokens or shuffle segments, such as "with" and "cancer may" are shuffled, which makes the augmented text no longer semantically coherent, and even modifies the semantic consistency of the text to affect the prediction of entity labels. The Text-to-Text based generative models cannot leverage flat $\langle B-Type \rangle$ and $\langle I-Type \rangle$ labels to mark the "stomach" token in the discontinuous entities: "stomach discomfort" and "stomach pain", thus limiting the application of this method to nested and discontinuous NER tasks.

To maintain text semantic coherence during augmentation and preserve entities for various NER tasks, in this work, we propose a novel **Entity-to-Text** instead of **Text-to-Text** based data augmentation approach named EnTDA. As illustrated in Figure 2, we first obtain the entity list [EU, German, British] in the original text, and then add, delete, swap, and replace the entity in the entity list to obtain the augmented entity list, e.g. [EU, German, British, Spanish]. We investigate that leveraging the rule-based methods to modify the entities in the entity list could generate more combinatorial entity lists without introducing grammatical errors. Then we adopt a conditional language model to generate the semantically coherent augmented text based on the augmented entity list. Thanks to the augmented entity list (including flat, nested, and discontinuous entities) we have already obtained, we can mark these preserved entities in the augmented text as shown in Figure 4. Since the augmented entity list provide the similar entity information in the text augmented by the language model, which may leads to insufficient diversity of text generation. Therefore, we propose a diversity beam search method for generative models to enhance text diversity. Overall, the main contributions of this work are as follows:

- To the best of our knowledge, we propose the first Entity-to-Text based data augmentation technique EnTDA. EnTDA leverages the pretrained large language model with semantic coherence and entity preserving to generate the augmented text, which could be used to benefit for all NER tasks (flat, nested, and discontinuous NER tasks).
- We propose the diversity beam search strategy for EnTDA to increase the diversity of the augmented text during generation process.
- We show that EnTDA outperforms strong data augmentation baselines across three NER tasks and two settings (full data and low resource settings).

	Techniques	Coher.	Diver.	NER Tasks		
				Flat	Nested	Discon.
Rule Based Techniques	Label-wise token rep.	–	–	✓	✓	✓
	Synonym replacement	–	–	✓	✓	✓
	Mention replacement	–	–	✓	✓	✗
	Shuffle within segments	–	–	✓	✓	✓
Generative Techniques	DAGA (Ding et al., 2020)	✓	–	✓	✗	✗
	MELM (Zhou et al., 2022)	✓	–	✓	✗	✗
	EnTDA	✓	✓	✓	✓	✓

Table 1: Comparison of different categories of techniques. "Coher." means "Semantic Coherence" and "Diver." means "Diveristy".

2 Related Work

2.1 Various NER Tasks

Named Entity Recognition (NER) is a pivotal task in IE which aims at locating and classifying named entities from texts into the predefined types such as `PERSON`, `LOCATION`, etc. (Chiu and Nichols, 2016; Xu et al., 2017; Yu et al., 2020). In addition to flat NER task (Sang and De Meulder, 2003), Kim et al. (2003) proposed nested NER task in the molecular biology domain. For example, in the text: *Alpha B2 proteins bound the PEBP2 site*, the entity *PEBP2* belongs to the type `PROTEIN` and *PEBP2 site* belongs to `DNA`.

Furthermore, some entities recognized in the text could be discontinuous (Mowery et al., 2013, 2014; Karimi et al., 2015). For example, in the text: *I experienced severe pain in my left shoulder and neck*, the entities *pain in shoulder* and *pain in neck* contain non-adjacent mentions. Some previous works proposed the unified frameworks which are capable of handling both three NER tasks (Li et al., 2020; Yan et al., 2021; Li et al., 2021). However, there is no unified data augmentation method designed for all three NER tasks due to the complexity of entity overlap. In this work, we try to bridge this gap and propose the first generative augmentation approach EnTDA that can be used to generate augmented data for all NER tasks (flat, nested, and discontinuous NER tasks).

2.2 Data Augmentation for NLP and NER

As shown in Table 1, we compare EnTDA with rule-based and traditional generative techniques, and present the comparison results below.

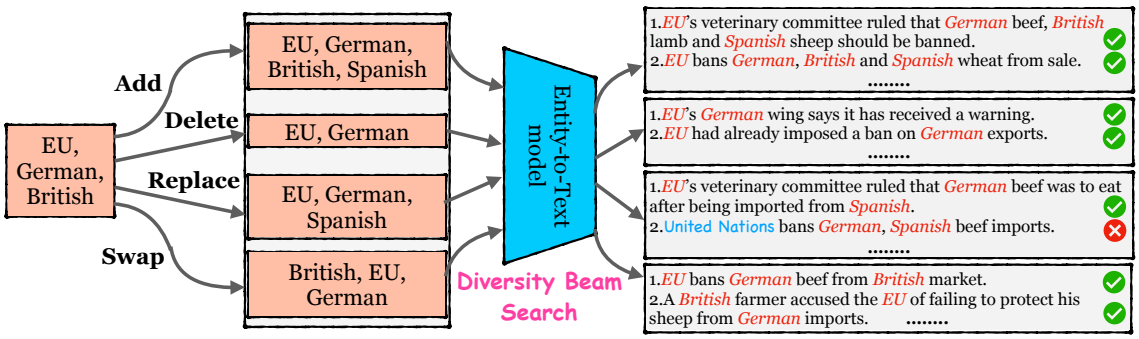


Figure 2: Overview of the proposed Entity-to-Text based data augmentation approach ENTDA. We first augment entity list via adding, deleting, replacing and swapping entities. Then the augmented entities will generate texts by adopting pretrained language model with diversity beam search. We finally mark the preserved entities in the augmented texts. Note that the texts that do not match the preserved entity list will be discarded.

Rule-based Augmentation Various rule-based augmentations for NLP tasks such as word replacement (Zhang et al., 2015; Cai et al., 2020), random deletion (Kobayashi, 2018; Wei and Zou, 2019), and word swapping (Şahin and Steedman, 2018; Min et al., 2020) manipulate the words in the original texts to generate synthetic texts. However, these manipulated tokens could not maintain the original labels since the change of syntax and semantics.

Dai and Adel (2020) proposes a replacement augmentation method to decide whether the selected token should be replaced by a binomial distribution, and if so, then the token will be replaced by another token with the same label. Furthermore, the similar approaches could be extended from token-level to mention-level. However, these methods still inevitably introduce incoherent replacement. In this work, we try to introduce the Entity-to-Text based augmentation approach to improve the coherence of the augmented texts.

Generative Augmentation Classic generative augmentations for NLP tasks such as back translation, which could be used to train a question answering model (Yu et al., 2018) or transfer texts from a high-resource language to a low-resource language (Hou et al., 2018; Xia et al., 2019). Anaby-Tavor et al. (2020); Kumar et al. (2020) adopt language model which is conditioned on sentence-level tags to modify original data for classification tasks exclusively. To utilize generative augmentation on more fine-grained and fragile token-level NER tasks, Ding et al. (2020) treats the NER labeling task as a text tagging task and requires generative models to annotate entities during generation. Zhou et al. (2022) builds the pretrained masked language models on corrupted training sentences and focuses on entity replacement. However, these methods rely on the Text-to-Text based generative models which cannot tag a token with nested labels during generation. In this work, we adopt the Entity-to-Text based generative model to tackle all NER tasks and bootstrap the diversity of the model with diversity beam search.

3 General NER Task Formulation

Considering that ENTDA has sufficient augmentation ability on flat, nested and discontinuous NER, we first formulate the general NER task framework as follows. Given an input text $X = [x_1, x_2, ..., x_n]$ of length n and the entity type set T, the output is an entity list $E = [e_1, e_2, ..., e_m, ..., e_l]$ of l entities, where $e_m = [s_{m1}, d_{m1}, ..., s_{mj}, d_{mj}, t_m]$. The s, d are the start and end indexes of a space in the text X. The j indicates that the entity consists of j spans. The t_m is an entity type in the entity type set T. For example, the discontinuous entity `stomach pain` in the text: "*The cancer patient has constant stomach discomfort and pain*" will be represented as $e_m = [5, 5, 8, 8, DISORDER]$.

4 Proposed Method

The proposed Entity-to-Text based data augmentation approach ENTDA consists of three modules: Entity List Augmentation, Entity-to-Text Generation, and Augmented Text Exploitation. Now we give the details of the three modules.

4.1 Entity List Augmentation

Entity List Augmentation aims to adopt four rule-based methods: Add, Delete, Replace, and Swap to modify the entities in the entity list obtained from the original sentences. Now, we give the

details of four operations on the original entity list $E = [e_1, e_2, ..., e_m, ..., e_l]$ as follows:

① **Add**. We first randomly select an entity e_m from the entity list E. Then we search for other entities in the training set and add $e_m^{'}$ with the same entity type as e_m to the original entity list: $E = [e_1, e_2, ..., e_m, e_m^{'}, ..., e_l]$.

② **Delete**. We randomly select an entity e_m from the original entity list E and delete it as $E = [e_1, e_2, ..., e_{m-1}, e_{m+1}, ..., e_l]$.

③ **Replace**. We first randomly select an entity e_m from the original entity list E. Similar to ①, we search $e_m^{'}$ with the same entity type to replace e_m as $E = [e_1, e_2, ..., e_m^{'}, ..., e_l]$.

④ **Swap**. We randomly select two entities e_m, $e_m^{'}$ in the original entity list E and swap their positions as $E = [e_1, e_2, ..., e_m^{'}, ..., e_m, ..., e_l]$.

4.2 Entity-to-Text Generation

After we obtain the augmented entity lists, the Entity-to-Text Generation module aims to generate the text for each entity list. Since the augmented entity list provide the similar entity information for augmented text, so we propose a diversity beam search method to increase text diversity.

Compared to traditional generation models that rely on greedy decoding (Chickering, 2002) and choosing the highest-probability logit at every generation step, we adopt a diversity beam search decoding strategy. More specifically, we first inject the entity types into the augmented entity list $E = [[t_1], e_1, [/t_1], ..., [t_m], e_m, [/t_m], ..., [t_l], e_l, [/t_l]]$ as the input sequence, which should provide sufficient type guidance for the generation model, then we adopt T5 (Raffel et al., 2020) as the generation model. We first fine-tune T5 on the original Entity-to-Text data and then adopt T5 (θ) to estimate the conditional probability distribution over all tokens in the dictionary $\mathcal{V}$ at time step t as:

$$\theta\left(y_t\right) = \log \Pr\left(y_t \mid y_{t-1}, \ldots, y_1, E\right). \quad (1)$$

where y_t is the t^{th} output token y in texts. We simplify the sum of log-probabilities (Eq. 1) of all previous tokens decoded $\Theta(\mathbf{y}_{[t]})$ as:

$$\Theta\left(\mathbf{y}_{[t]}\right) = \sum_{\tau \in [t]} \theta\left(y_\tau\right), \quad (2)$$

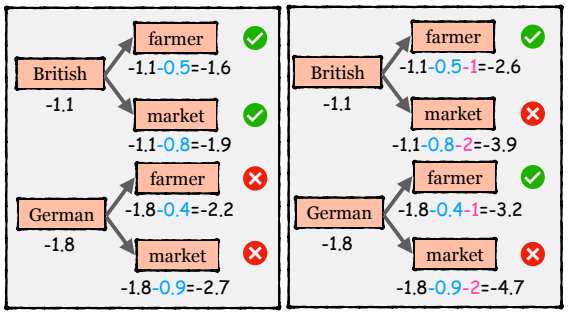


Figure 3: The example about the generated text with beam search decoding (left) and diversity beam search decoding (right). The hyperparameter γ is set to 1 and beam width B is set to 2 here.

where $\mathbf{y}_{[t]}$ is the token list consisting of $[y_1, y_2, ..., y_t]$. Therefore, our decoding problem is transformed into the task of finding the text that could maximize $\Theta(\mathbf{y})$. The classical approximate decoding method is the beam search (Wiseman and Rush, 2016), which stores top beam width B candidate tokens at time step t. Specifically, beam search selects the B most likely tokens from the set:

$$\mathcal{Y}_t = Y_{[t-1]} \times \mathcal{V}, \quad (3)$$

where $Y_{[t-1]} = \left\{\mathbf{y}_{1,[t-1]}, \ldots, \mathbf{y}_{B,[t-1]}\right\}$ and $\mathcal{V}$ is the dictionary. However, traditional beam search keeps a small proportion of candidates in the search space and generates the texts with minor perturbations (Huang, 2008), which impedes the diversity of generated texts. Inspired by Vijayakumar et al. (2016), we introduce an objective to increase the dissimilarities between candidate texts and finalize the Eq. 2 as diversity beam search decoding:

$$\hat{\Theta}\left(\mathbf{y}_{[t]}\right) = \sum_{\tau \in [t]} \left(\theta\left(y_\tau\right) - \gamma k_\tau\right), \quad (4)$$

where γ is a hyperparameter and represents the punishment degree. k_τ denotes the ranking of the current tokens among candidates. In practice, it's a penalty text of beam width: $[1, 2, ..., B]$ which punishes bottom ranked tokens among candidates and thus generates tokens from diverse previous tokens. For a better understanding, we give an example about the text with beam search decoding and diversity beam search decoding in Figure 3.

The traditional greedy decoding chooses the highest-probability logit at every generation step and results in *British farmer*. Compared to the diversity beam search decoding method, the beam search decoding method maintains a small proportion of candidates in the search space without

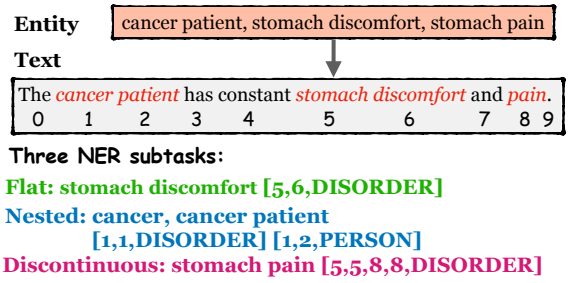


Figure 4: Details of marking the texts with the augmented entity lists for three NER tasks.

the introduction of a penalty text of beam width $[1, 2, ..., B]$. This additional objective increases the dissimilarities between candidate texts and thus generates tokens from diverse previous tokens. For example, *British farmer* and *German farmer* are generated instead of *British farmer* and *British market*, which brings the diversity token *German*. Likewise, the diversity token *market* will also be considered in the subsequent generation. Overall, at each time step t:

$$Y_{[t]} = \underset{\mathbf{y}_{1,[t]},\cdots,\mathbf{y}_{B,[t]} \in \mathcal{Y}_t}{\operatorname{argmax}} \sum_{b \in [B]} \hat{\Theta}\left(\mathbf{y}_{b,[t]}\right). \quad (5)$$

This process will generate the most likely texts that are selected by ranked the B beams based on the diversity beam search decoding.

4.3 Augmented Text Exploitation

To utilize these augmented Entity-to-Text data, we need to mark the texts with the augmented entity lists. As illustrated in Figure 2, we first automatically judge whether the entities match the tokens in the texts and remove these noisy texts of mismatched entities. For example, *EU* is generated as *United Nations* and this generated text is automatically deleted. Then as illustrated in Figure 4, we provide the details of the text marking process:

(1) If the entity is **flat**, we obtain the start and end position indexes through the exact match between entity and text.

(2) If the entity is **nested**, we first store all the overlapping entity mentions belonging to the same nested entity and match these mentions with text to obtain start and end position indexes.

(3) If the entity is **discontinuous**, we match the entity mentions which belong to the same discontinuous entity with text to obtain start and end position indexes.

Note that the process of text marking is done automatically based on the above three situations. After we obtain these augmented data with marked flat, nested, and discontinuous entities, we naturally formulate the texts as input to NER tasks.

5 Experiments and Analyses

We conduct extensive experiments on thirteen NER datasets across three tasks (flat, nested, and discontinuous NER) and two settings (full data and low resource NER) to show the effectiveness of EnTDA on NER, and give a detailed analysis.

5.1 Backbone Models

We adopt two SOTA backbone models which could solve all three NER tasks:

1) **The unified Seq2Seq framework** (Yan et al., 2021) formulates three NER tasks as an entity span text generation task without the special design of the tagging schema to enumerate spans.

2) **The unified Word-Word framework** (Li et al., 2022b) models the neighboring relations between entity words as a 2D grid and then adopts multi-granularity 2D convolutions for refining the grid representations.

These two backbone models are leveraged to solve the general NER tasks illustrated in Section 3 and demonstrate the effectiveness of EnTDA.

5.2 Datasets

To demonstrate that EnTDA could be used in various NER tasks and backbone models, we follow Yan et al. (2021); Li et al. (2022b) and adopt the same datasets (split) as follows:

1) **Flat NER Datasets**: We adopt the CoNLL-2003 (Sang and De Meulder, 2003) and OntoNotes (Pradhan et al., 2013) datasets. For OntoNotes, we evaluate in the English corpus with the same setting as Yan et al. (2021).

2) **Nested NER Datasets**: We adopt the ACE 2004 (Doddington et al., 2004), ACE 2005 (Christopher Walker and Maeda., 2005) and GENIA (Kim et al., 2003) datasets. Following Yan et al. (2021), we split the ACE 2004/ACE 2005 into train/dev/test sets by 80%/10%/10% and GENIA into 81%/9%/10% respectively.

3) **Discontinuous NER Datasets** We adopt the CADEC (Karimi et al., 2015), ShARe13 (Mowery et al., 2013) and ShARe14 (Mowery et al., 2014) datasets from biomedical domain. Following Yan et al. (2021), we split the CADEC into train/dev/test sets by 70%/15%/15% and use 10% training set as the development set for ShARe13/ShARe14.

Method / Datasets	Flat NER datasets		Nested NER datasets			Discontinuous NER datasets			AVG.	Δ
	CoNLL2003	OntoNotes	ACE2004	ACE2005	Genia	CADEC	ShARe13	ShARe14		
Unified Word-Word Framework	93.14	90.66	87.54	86.72	81.34	73.22	82.57	81.79	84.62	–
+Label-wise token rep.	93.32	90.78	87.83	86.98	81.65	73.47	82.84	82.07	84.87	0.25↑
+Synonym replacement	93.35	90.75	87.87	86.93	81.63	73.50	82.87	82.10	84.88	0.26↑
+Mention replacement	93.29	90.80	87.89	86.97	81.64	–	–	–	–	–
+Shuffle within segments	93.30	90.68	87.68	86.84	81.47	73.36	82.71	81.92	84.75	0.13↑
+DAGA	93.47	90.89	–	–	–	–	–	–	–	–
+MELM	93.60	91.06	–	–	–	–	–	–	–	–
+**EnTDA (Delete)**	93.82	91.23	**88.29**	87.54	82.12	73.86	83.31	82.45	85.33	0.71↑
+**EnTDA (Add)**	**93.93**	91.26	88.27	**87.60**	82.19	**73.89**	83.34	**82.55**	**85.42**	**0.76**↑
+**EnTDA (Replace)**	93.87	91.21	88.18	87.46	**82.40**	73.82	83.19	82.52	85.33	0.71↑
+**EnTDA (Swap)**	93.91	91.25	88.18	87.54	82.32	73.81	83.30	82.52	85.35	0.73↑
+**EnTDA (All)**	93.88	**91.34**	88.21	87.56	82.25	73.86	**83.35**	82.47	85.37	0.75↑
+**EnTDA (None)**	93.44	90.89	87.84	87.01	81.73	73.57	82.90	82.09	84.93	0.31↑
+**EnTDA (All) w/o Diver.**	93.55	91.01	87.93	87.23	81.91	73.75	83.02	82.20	85.08	0.46↑
Unified Seq2Seq Framework	92.78	89.51	86.19	84.74	79.10	70.76	79.69	79.40	82.78	–
+Label-wise token rep.	92.91	89.68	86.33	85.04	79.41	71.22	79.93	79.64	83.03	0.25↑
+Synonym replacement	92.85	89.59	86.28	85.32	79.36	71.18	79.86	79.55	83.00	0.22↑
+Mention replacement	92.80	89.80	86.14	85.01	79.44	–	–	–	–	–
+Shuffle within segments	92.85	89.40	86.22	84.99	79.28	71.13	79.72	79.50	82.89	0.11↑
+DAGA	92.92	89.97	–	–	–	–	–	–	–	–
+MELM	92.95	89.95	–	–	–	–	–	–	–	–
+**EnTDA (Delete)**	93.38	90.23	86.51	86.26	80.80	71.51	80.58	80.04	83.67	0.89↑
+**EnTDA (Add)**	93.27	90.27	86.73	86.39	80.88	71.50	**80.92**	80.16	83.77	0.99↑
+**EnTDA (Replace)**	93.32	90.16	86.55	**86.41**	80.74	71.64	80.64	80.23	83.71	0.93↑
+**EnTDA (Swap)**	93.45	90.04	86.40	86.30	80.67	71.37	80.37	80.12	83.59	0.81↑
+**EnTDA (All)**	**93.51**	**90.31**	**86.92**	86.39	**80.94**	**71.70**	80.83	**80.36**	**83.87**	**1.09**↑
+**EnTDA (None)**	92.90	90.02	86.28	85.57	79.66	71.30	80.13	79.71	83.20	0.42↑
+**EnTDA (All) w/o Diver.**	93.13	90.21	86.47	85.78	79.88	71.54	80.31	79.97	83.41	0.63↑

Table 2: F1 results of various NER tasks. For all three backbone models and six baseline augmentation approaches, we rerun their open source code and adopt the given parameters.

We show the detailed statistics and entity types of the datasets in Appendix A.

5.3 Baseline Augmentation Methods

Unlike sentence-level classification tasks, NER is a fine-grained token-level task, so we adopt six entity-level data augmentation baselines, which are designed for various NER tasks.

The four rule-based baseline augmentation techniques: (1) **Label-wise token replacement** (Dai and Adel, 2020) utilizes a binomial distribution to decide whether each token should be replaced, and then replaces the chosen token with another token that has the same entity type. (2) **Synonym replacement** (Dai and Adel, 2020) replaces the chosen token with the synonym retrieved from WordNet. (3) **Mention replacement** (Dai and Adel, 2020) replaces the chosen entity with another entity, which has the same entity type. (4) **Shuffle within segments** (Dai and Adel, 2020) splits the sentences into segments based on whether they come from the same entity type, and uses a binomial distribution to decide whether to shuffle tokens within the same segment. The two generative baseline augmentation techniques are: (5) **DAGA** (Ding et al., 2020) treats the NER labeling task as a text tagging task and annotates entities with generative models during generation. (6) **MELM** (Zhou et al., 2022) generates augmented data with diverse entities, which is built upon pre-trained masked language models. MELM is further finetuned on corrupted training sentences with only entity tokens being randomly masked to focus on entity replacement.

We present another model: **EnTDA (All)**, which adopts four entity list operations simultaneously to generate augmented texts. Note that we focus on entity-level NER augmentation tasks, so to the best of our knowledge, we have employed all entity-level augmentation techniques.

5.4 Experiment Settings

For EnTDA, we fine-tune the T5-Base (Raffel et al., 2020) with the initial parameters on the Entity-to-Text data of the training set and utilize the default tokenizer with max-length as 512 to preprocess the data. We use AdamW (Loshchilov and Hutter, 2018) with $5e{-}5$ learning rate to optimize the cross entropy loss. The batch size is set to 5 and the number of training epoch is set to 3. During diversity beam search decoding, we set γ as 10 and beam width B as 3, which means that each entity set will generate three texts.

EnTDA and all baselines augment the training set by 3x for a fair comparison. For example, the number of texts in the training set is 100, we generate 300 texts and add them to the training set. We replace the language model in MELM (Zhou et al., 2022) with XLM-RoBERTa-large (355M) (Conneau et al., 2020), and we use T5-Base (220M) with fewer parameters for comparison.

5.5 Results and Analyses

Table 2 shows the average F1 results on three runs. All backbone NER models gain F1 performance improvements from the augmented data when compared with the models that only use original training data, demonstrating the effectiveness of data augmentation approaches in the various NER tasks. Surprisingly, EnTDA (None) outperforms the baseline methods by 0.11% F1 performance among the backbone models, which shows that the generative models using a diversity beam search have sufficient capacity to generate high-quality augmented data.

More specifically, for flat NER datasets, MELM is considered as the previous SOTA data augmentation approach. The proposed EnTDA (All) on average achieves 0.23% higher in F1 among flat NER datasets and two backbone models. For nested and discontinuous NER datasets, the label-wise token replacement method achieves the best performance among baselines. EnTDA (All) achieve an average 0.78% F1 boost among nested and discontinuous NER datasets, which demonstrates that leveraging generative model to augment semantically coherent texts is effective.

Among all NER datasets, EnTDA is undoubtedly capable of achieving state-of-the-art results (with student's T test $p < 0.05$). Except EnTDA (All), EnTDA (Add) achieves the largest F1 performance gains of 0.99% and 0.76% on the unified Seq2Seq and Word-Word frameworks, respectively. We attribute this delightful improvement of the "Add" operation to the additionally introduced knowledge: we add the entity from the training set with the same entity type.

Ablation Study

In Table 2, we remove the entity list augmentation module (EnTDA(None)), or change the diversity beam search to the traditional beam search (EnTDA(All) w/o Diver.). We can conclude that entity list augmentation and diversity beam search modules bring an average F1 improvement of 0.56% and 0.38% on the eight datasets. Using the entity list augmentation module can give a richer entity combination, which brings more improvement. Adopting the diversity beam search brings more diverse texts and gains greater improvements.

Method / Datasets	CoNLL2003	ACE2005	CADEC
Unified Word-Word Framework	86.83	79.56	65.03
+Label-wise token rep.	87.23	79.97	65.50
+Synonym replacement	87.16	80.01	65.46
+Mention replacement	87.30	80.10	–
+Shuffle within segments	87.04	79.85	65.28
+DAGA	87.82	–	–
+MELM	88.24	–	–
+EnTDA (Delete)	89.91	81.94	69.12
+EnTDA (Add)	90.13	**82.15**	69.03
+EnTDA (Replace)	90.07	82.01	69.29
+EnTDA (Swap)	89.97	81.98	69.25
+EnTDA (All)	**90.22**	82.08	**69.31**
Unified Seq2Seq Framework	85.90	77.32	62.24
+Label-wise token rep.	86.44	77.81	62.56
+Synonym replacement	86.73	77.79	62.61
+Mention replacement	86.94	77.83	–
+Shuffle within segments	86.26	77.65	62.49
+DAGA	87.05	–	–
+MELM	87.43	–	–
+EnTDA (Delete)	89.20	79.10	66.04
+EnTDA (Add)	89.62	79.23	**66.42**
+EnTDA (Replace)	89.41	79.02	66.21
+EnTDA (Swap)	88.96	78.96	65.93
+EnTDA (All)	**89.82**	**79.51**	66.40

Table 3: F1 results of various NER tasks under low resource scenarios.

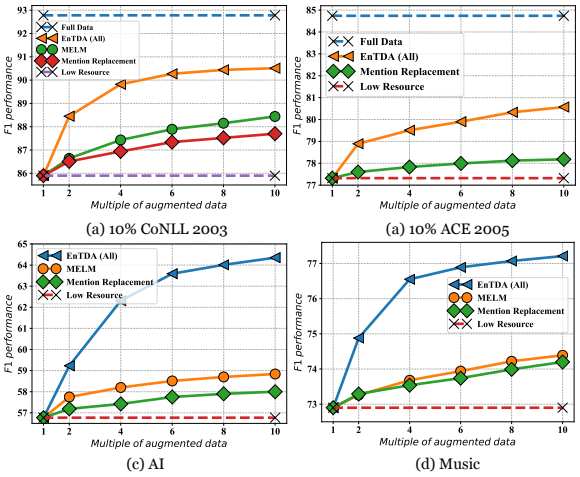
Handling Low Resource NER Scenarios

We further introduce an extreme yet practical scenario: only limited labeled data is available. This low resource NER scenario demonstrates that our EnTDA approach bootstraps the generalization ability of the NER model and is a quite appealing approach for data-oriented applications in the real-world. In practice, we randomly choose 10% training data from CoNLL2003/ACE2005/CADEC to represent the three NER tasks. Note that the fine-tuning of T5-large and our four operations on the entity list are also done on 10% training data.

From Table 3, compared to training directly on the 10% training set, leveraging the augmented data achieves the performance improvement in F1. We also observe that EnTDA approach obtains the most competitive F1 performance improvement when compared with baseline data augmentation approaches. More specifically, EnTDA (All) achieve an average 2.97% F1 boost among three backbone models, which means EnTDA obtains more performance gains under the low resource scenario than in the full data scenario. Especially for the most challenging discontinuous dataset CADEC, EnTDA (All) obtains the largest F1 performance gain of 4.22%. Surprisingly, on

Method / Datasets	Politics	Natural Science	Music	Literature	AI
Seq2Seq Framework	70.11	70.72	72.90	63.69	56.77
+Label-wise token rep.	70.45	70.91	73.48	63.97	57.04
+Synonym replacement	70.43	71.04	73.66	63.92	57.34
+Mention replacement	70.47	71.07	73.54	64.02	57.42
+Shuffle within segments	70.39	70.94	73.30	63.88	57.26
+DAGA	71.06	71.51	73.46	64.21	57.83
+EnTDA (Delete)	72.60	72.05	75.87	67.18	61.58
+EnTDA (Add)	72.81	**72.55**	76.20	67.82	61.97
+EnTDA (Replace)	72.94	72.46	76.12	67.57	61.89
+EnTDA (Swap)	72.47	71.89	75.58	67.06	61.37
+EnTDA (All)	**72.98**	72.47	**76.55**	**68.04**	**62.31**

Table 4: F1 results of real low resource NER tasks.



(a) 10% CoNLL 2003 (a) 10% ACE 2005

(c) AI (d) Music

Figure 5: F1 results of the unified Seq2Seq framework with augmented data at various multiples on different low resource datasets.

10% CoNLL2003, EnTDA (All) has only a 2.94% decrease in F1 performance compared to using the full training data, but EnTDA (All) saves 10x the annotated data, which shows that adopting EnTDA is quite appealing for real-world applications.

Tackling Real Low Resource NER Tasks

We adopt real low resource NER datasets (Liu et al., 2021) from Wikipedia which contains politics, natural science, music, literature and artificial intelligence domains with only 100 or 200 labeled texts in the training set. EnTDA and baseline data augmentation approaches still augment the training set by 3x. From Table 4, we are delighted to observe EnTDA could quickly learn from the extremely limited Entity-to-Text data and bring 3.45% F1 performance gains over various domains. Compared with baseline augmentation methods, EnTDA generates more diverse texts and undoubtedly gains greater advantages.

Various Augmentation Multiples Performance

We further vary the multiples of augmented data from 2x to 10x the training set to study the influence of data augmentation approaches for the NER backbone models under low resource scenarios. We choose different low resource datasets

Method / Datasets	CoNLL2003	CADEC	AI
Label-wise token rep.	8.12	8.87	7.52
Synonym replacement	7.44	7.88	7.01
Mention replacement	7.07	7.42	6.54
Shuffle within segments	10.24	12.32	9.65
DAGA	5.46	6.23	5.07
MELM	5.27	6.29	4.82
EnTDA (All)	**4.74**	**5.19**	**4.28**

Table 5: Perplexity of the augmented data with various augmentation approaches. Lower perplexity is better.

Methods / Datasets	CoNLL2003		CADEC		AI	
	TTR	Diver.	TTR	Diver.	TTR	Diver.
Label-wise token rep.	81.2	3.1	80.5	3.4	81.9	3.3
Synonym replacement	81.9	3.3	80.1	3.5	82.6	3.4
Mention replacement	83.8	3.9	82.9	3.6	**84.2**	3.8
Shuffle within segments	72.9	2.4	71.6	2.0	73.7	2.1
DAGA	73.8	2.8	74.1	2.6	74.3	3.1
MELM	77.2	3.2	78.1	2.9	76.6	3.0
EnTDA (All)	**86.4**	**4.3**	**85.1**	**4.5**	83.7	**4.4**

Table 6: Diversity Evaluation on three datasets.

and three representative augmentation approaches (Mention replacement, MELM, and EnTDA (All)), then represent the results in Figure 5.

We could observe that the unified Seq2Seq framework has more performance gains with ever-increasing augmented data. EnTDA (All) consistently achieves better F1 performance, with a clear margin, compared to baseline augmentation approaches under various augmentation multiples. Especially for Music, EnTDA (All) brings an incredible 4.01% improvement in F1 performance with only 300 augmented data.

Semantic Coherence Analysis

Compared with baseline augmentation approaches, EnTDA conditionally generates texts with the diversity beam search decoding, which provides more coherent texts. We analyze the coherence through perplexity based on a large Transformer language model: GPT-2 (Radford et al., 2019). From Table 5, EnTDA obtains the lowest perplexity. Although DAGA and MELM are also based on generative models, the texts are not natural enough since only partial text is replaced.

Diversity Evaluation

We measure the diversity of augmented sentences through automatic and manual metrics. For automatic metric, we introduce the Type-Token Ratio (TTR) (Tweedie and Baayen, 1998) to evaluate the ratio of the number of different words to the total number for each original text. Higher TTR (%) indicates more diversity in sentences. Besides that, we ask 5 annotators to give a score for the degree of diversity of the 200 generated texts, with

Approach: Original Sentence Entity: **unsupervised learning, principal component analysis , cluster analysis** Entity Type: **field, algorithm, algorithm** Text: The main methods used in *unsupervised learning* are *principal component analysis* and *cluster analysis*.
Approach: ENTDA (Add) Entity: **unsupervised learning, principal component analysis , cluster analysis, dimension reduction** Entity Type: **field, algorithm, algorithm, algorithm** Text: In *unsupervised learning*, *principal component analysis*, *cluster analysis* and *dimension reduction* are used to reduce the number of variables in a task.
Approach: ENTDA (Delete) Entity: **unsupervised learning, principal component analysis** Entity Type: **field, algorithm** Text: In the field of *unsupervised learning*, *principal component analysis* is used to model the learning process.
Operation: ENTDA (Replace) Entity: **unsupervised learning, principal component analysis , dimension reduction** Entity Type: **field, algorithm, algorithm** Text: In the field of *unsupervised learning*, *principal component analysis* and *dimension reduction* are used to reduce the size of the data.
Operation: ENTDA (Swap) Entity: **unsupervised learning, cluster analysis , principal component analysis** Entity Type: **field, algorithm, algorithm** Text: *Unsupervised learning* uses *cluster analysis* and *principal component analysis* to learn a task.
Operation: ENTDA (All) Entity: **unsupervised learning, dimension reduction , principal component analysis** Entity Type: **field, algorithm, algorithm** Text: *Unsupervised learning* uses *cluster analysis* to achieve the purpose of *dimension reduction* for better learning a task.
Approach: Mention Replacement Entity: **heterodyning, principal component analysis , cluster analysis** Entity Type: **field, algorithm, algorithm** Text: The main methods used in *heterodyning* are *principal component analysis* and *cluster analysis*.
Operation: DAGA Text: *Unsupervised learning* uses *principal component analysis* and *cluster analysis*. Entity (Unchanged): **unsupervised learning, principal component analysis, cluster analysis** Entity Type (Unchanged): **field, algorithm, algorithm**

Table 7: The augmented texts for AI domain. We show six approaches to generate texts marked with the corresponding *entity* list.

score range of 1~5. According to the annotation guideline in Appendix D, a higher score indicates the method can generate more diverse texts.

We present the average scores on the datasets in Table 6. ENTDA could obtain 7.8% TTR and 1.4 diversity performance boost in average compared to MELM.

6 Case Study

We show eight approaches to obtain augmented data for the AI domain in Table 7. Compared with baseline augmentation methods, ENTDA introduces a knowledge expansion and conditionally generates texts based on the diversity beam search, which provides more coherent and diverse texts. For example, The Mention Replacement approach replaces the entities `unsupervised learning` with `heterodyning`, which ignores the semantics of the context and makes an ungrammatical replacement, resulting in incoherent and unreasonable texts. For the DAGA approach, it simply stacks three entities: `unsupervised learning, principal component analysis, cluster analysis` in the text, which could not provide knowledge expansions to the NER models.

7 Conclusions and Future Work

In this paper, we propose an Entity-to-Text based data augmentation approach ENTDA for NER tasks. Compared with traditional rule-based augmentation methods that break semantic coherence, or use Text-to-Text based augmentation methods that cannot be used on nested and discontinuous NER tasks, our method can generate semantically coherent texts for all NER tasks, and use the diversity beam search to improve the diversity of augmented texts. Experiments on thirteen public real-world datasets, and coherence and diversity analysis show the effectiveness of ENTDA. Moreover, we can also apply the method of data augmentation to low-resource relation extraction (Hu et al., 2020, 2021b,a; Liu et al., 2022b; Hu et al., 2023), natural language inference (Li et al., 2023, 2022c), semantic parsing (Liu et al., 2022a, 2023), and other NLP application tasks, thus realizing knowledge enhancement based on data augmentation approach.

8 Limitations

We discuss the limitations of our method from three perspectives.

First, our method is based on pre-trained language models, so compared to rule-based data augmentation methods (synonym replacement, shuffle within segments, etc.), our method requires higher time complexity.

Second, the entity matching process (Section 4.3) will discard sentences which cannot match entities in the entity list, which will affect the utilization of data.

Third, our data augmentation method based on the pre-trained language models, whose generalization ability is limited since the augmented knowledge comes from the pre-trained language models. However, the knowledge in pre-trained language models is limited and not domain-specific. How to improve the generalization ability of the data augmentation methods is a future research work.

9 Acknowledgement

We thank the reviewers for their valuable comments. Yong Jiang and Lijie Wen are the corresponding authors. Xuming Hu, Aiwei Liu and Lijie Wen were partially supported by the National Key Research and Development Program of China (No. 2019YFB1704003), the National Nature Science Foundation of China (No. 62021002), Tsinghua BNRist and Beijing Key Laboratory of Industrial Bigdata System and Application. Philip S. Yu was partially supported by the NSF under grants III-1763325, III-1909323, III-2106758, SaTC-1930941.

References

Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. 2020. Do not have enough data? deep learning to the rescue! In *Proc. of AAAI*, volume 34, pages 7383–7390.

Hengyi Cai, Hongshen Chen, Yonghao Song, Cheng Zhang, Xiaofang Zhao, and Dawei Yin. 2020. Data manipulation: Towards effective instance learning for neural dialogue generation via learning to augment and reweight. In *Proc. of ACL*, pages 6334–6343.

David Maxwell Chickering. 2002. Optimal structure identification with greedy search. *Journal of machine learning research*, 3(Nov):507–554.

Jason PC Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns. *TACL*, 4:357–370.

Julie Medero Christopher Walker and Kazuaki Maeda. 2005. Ace 2005 multilingual training corpus. In *Linguistic Data Consortium, Philadelphia 57*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proc. of ACL*, pages 8440–8451.

Xiang Dai and Heike Adel. 2020. An analysis of simple data augmentation for named entity recognition. In *Proc. of COLING*, pages 3861–3867, Barcelona, Spain (Online). International Committee on Computational Linguistics.

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

George R Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. The automatic content extraction (ace) program–tasks, data, and evaluation. In *Proc. of LREC*.

Steven Y Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for nlp. In *Proc. of ACL-IJCNLP: Findings*, pages 968–988.

Yutai Hou, Yijia Liu, Wanxiang Che, and Ting Liu. 2018. Sequence-to-sequence data augmentation for dialogue language understanding. In *Proc. of COLING*, pages 1234–1245.

Xuming Hu, Zhaochen Hong, Chenwei Zhang, Irwin King, and Philip S Yu. 2023. Think rationally about what you see: Continuous rationale extraction for relation extraction. *arXiv preprint arXiv:2305.03503*.

Xuming Hu, Lijie Wen, Yusong Xu, Chenwei Zhang, and Philip S. Yu. 2020. Selfore: Self-supervised relational feature learning for open relation extraction. In *Proc. of EMNLP*, pages 3673–3682.

Xuming Hu, Chenwei Zhang, Fukun Ma, Chenyao Liu, Lijie Wen, and Philip S. Yu. 2021a. Semi-supervised relation extraction via incremental meta self-training. In *Findings of EMNLP*, pages 487–496.

Xuming Hu, Chenwei Zhang, Yawen Yang, Xiaohe Li, Li Lin, Lijie Wen, and Philip S. Yu. 2021b. Gradient imitation reinforcement learning for low resource relation extraction. In *Proc. of EMNLP*, pages 2737–2746.

Liang Huang. 2008. Forest reranking: Discriminative parsing with non-local features. In *Proceedings of ACL-08: HLT*, pages 586–594.

Sarvnaz Karimi, Alejandro Metke-Jimenez, Madonna Kemp, and Chen Wang. 2015. Cadec: A corpus of adverse drug event annotations. *Journal of biomedical informatics*, 55:73–81.

J-D Kim, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii. 2003. Genia corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl_1):i180–i182.

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proc. of NAACL-HLT*.

Varun Kumar, Ashutosh Choudhary, and Eunah Cho. 2020. Data augmentation using pre-trained transformer models. In *Proceedings of the 2nd Workshop on Life-long Learning for Spoken Language Systems*, pages 18–26.

Bohan Li, Yutai Hou, and Wanxiang Che. 2022a. Data augmentation approaches in natural language processing: A survey. *AI Open*.

Fei Li, ZhiChao Lin, Meishan Zhang, and Donghong Ji. 2021. A span-based model for joint overlapped and discontinuous named entity recognition. In *Proc. of ACL-IJCNLP*, pages 4814–4828, Online. Association for Computational Linguistics.

Jingye Li, Hao Fei, Jiang Liu, Shengqiong Wu, Meishan Zhang, Chong Teng, Donghong Ji, and Fei Li. 2022b. Unified named entity recognition as word-word relation classification. In *Proc. of AAAI*, volume 36, pages 10965–10973.

Shuang Li, Xuming Hu, Li Lin, Aiwei Liu, Lijie Wen, and Philip S. Yu. 2023. A multi-level supervised contrastive learning framework for low-resource natural language inference. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:1771–1783.

Shu'ang Li, Xuming Hu, Li Lin, and Lijie Wen. 2022c. Pair-level supervised contrastive learning for natural language inference. *arXiv preprint arXiv:2201.10927*.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020. A unified MRC framework for named entity recognition. In *Proc. of ACL*, pages 5849–5859, Online. Association for Computational Linguistics.

Aiwei Liu, Xuming Hu, Li Lin, and Lijie Wen. 2022a. Semantic enhanced text-to-sql parsing via iteratively learning schema linking graph. In *Proc. of KDD*, pages 1021–1030.

Aiwei Liu, Xuming Hu, Lijie Wen, and Philip S Yu. 2023. A comprehensive evaluation of chatgpt's zero-shot text-to-sql capability. *arXiv preprint arXiv:2303.13547*.

Shuliang Liu, Xuming Hu, Chenwei Zhang, Shu'ang Li, Lijie Wen, and Philip S. Yu. 2022b. Hiure: Hierarchical exemplar contrastive learning for unsupervised relation extraction. In *Proc. of NAACL-HLT*, pages 5970–5980.

Zihan Liu, Yan Xu, Tiezheng Yu, Wenliang Dai, Ziwei Ji, Samuel Cahyawijaya, Andrea Madotto, and Pascale Fung. 2021. Crossner: Evaluating cross-domain named entity recognition. In *Proc. of AAAI*, volume 35, pages 13452–13460.

Ilya Loshchilov and Frank Hutter. 2018. Fixing weight decay regularization in adam.

Junghyun Min, R Thomas McCoy, Dipanjan Das, Emily Pitler, and Tal Linzen. 2020. Syntactic data augmentation increases robustness to inference heuristics. In *Proc. of ACL*, pages 2339–2352.

Danielle L. Mowery, Sumithra Velupillai, Brett R. South, Lee M. Christensen, David Martínez, Liadh Kelly, Lorraine Goeuriot, Noémie Elhadad, Sameer Pradhan, Guergana K. Savova, and Wendy W. Chapman. 2013. Task 1: Share/clef ehealth evaluation lab 2013. In *CLEF*.

Danielle L. Mowery, Sumithra Velupillai, Brett R. South, Lee M. Christensen, David Martínez, Liadh Kelly, Lorraine Goeuriot, Noémie Elhadad, Sameer Pradhan, Guergana K. Savova, and Wendy W. Chapman. 2014. Task 2: Share/clef ehealth evaluation lab 2014. In *CLEF*.

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 *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

Gözde Gül Şahin and Mark Steedman. 2018. Data augmentation via dependency tree morphing for low-resource languages. In *Proc. of EMNLP*, pages 5004–5009.

Erik Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proc. of HLT-NAACL*, pages 142–147.

Connor Shorten and Taghi M Khoshgoftaar. 2019. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):1–48.

Fiona J Tweedie and R Harald Baayen. 1998. How variable may a constant be? measures of lexical richness in perspective. *Computers and the Humanities*, 32(5):323–352.

Ashwin K Vijayakumar, Michael Cogswell, Ramprasaath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2016. Diverse beam search: Decoding diverse solutions from neural sequence models.

Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proc. of EMNLP-IJCNLP*, pages 6382–6388.

Sam Wiseman and Alexander M Rush. 2016. Sequence-to-sequence learning as beam-search optimization. In *Proc. of EMNLP*, pages 1296–1306.

Mengzhou Xia, Xiang Kong, Antonios Anastasopoulos, and Graham Neubig. 2019. Generalized data augmentation for low-resource translation. In *Proc. of ACL*, pages 5786–5796.

Mingbin Xu, Hui Jiang, and Sedtawut Watcharawittayakul. 2017. A local detection approach for named entity recognition and mention detection. In *Proc. of ACL*, pages 1237–1247.

Hang Yan, Tao Gui, Junqi Dai, Qipeng Guo, Zheng Zhang, and Xipeng Qiu. 2021. A unified generative framework for various NER subtasks. In *Proc. of ACL*, pages 5808–5822, Online. Association for Computational Linguistics.

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. In *ICLR*.

Juntao Yu, Bernd Bohnet, and Massimo Poesio. 2020. Named entity recognition as dependency parsing. In *Proc. of ACL*, pages 6470–6476.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *NeurIPS*, 28:649–657.

Ran Zhou, Xin Li, Ruidan He, Lidong Bing, Erik Cambria, Luo Si, and Chunyan Miao. 2022. Melm: Data augmentation with masked entity language modeling for low-resource ner. In *Proc. of ACL*, pages 2251–2262.

A Dataset Statistics

we show the detailed statistics of the datasets in Table 8. We further give details on entity types for thirteen datasets in Table 9.

B Entity Addition and Replacement Strategy

EnTDA add and replace the entity in the training set that has the same entity type. This strategy can provide the knowledge expansion during the generation, which is an appealing property when the hand-craft knowledge base is difficult to construct for augmentation approaches.

If we directly replace the entities in the text with other entities of the same type, this is equivalent to the baseline: Mention Replacement. From Table 2, 3 and 4, we could observe that compared to EnTDA (Replace), the improvement of F1 performance is greatly reduced. The main reason is that context-free entities are replaced, resulting in obscure and unreasonable texts. For example, "*EU's German wing says it has received a warning*." may be changed to "*EU's World War Two wing says it has received a warning*." since the two entities share the same type: MISC.

C Hyperparameter Analysis

We study the hyperparameter γ in the diversity beam search, which represents the degree of probability penalty in the decoding process and determines the diversity of sentences. Modifying γ allows us to control the diversity of the texts. We vary the γ from 1 to 100 and represent the F1 results using the unified Seq2Seq framework and EnTDA (All) in Table 10. With no more than 1% F1 fluctuating results among three datasets, EnTDA appears robust to the choice of γ.

D Annotation Guideline

Each annotator needs to carefully read each augmented text, compare it with the original text, and give a score according to the following criteria. Note that all augmented texts for a dataset are given an average score.

- Score:1. The augmented texts under the same original text are almost the same.
- Score:2. The augmented texts under the same original text are slightly different, with serious grammatical errors.
- Score:3. The augmented texts under the same original text are slightly different, and there are almost no grammatical errors.
- Score:4. The augmented texts under the same original text are diverse, with serious grammatical errors.
- Score:5. The augmented texts under the same original text are diverse, and there are almost no grammatical errors.

		Sentence					Entity			
		#All	#Train	#Dev	#Test	#Avg.Len	#All	#Nes.	#Dis.	#Avg.Len
Flat NER	CoNLL2003	20,744	17,291	–	3,453	14.38	35,089	–	–	1.45
	OntoNotes	76,714	59,924	8,528	8,262	18.11	104,151	–	–	1.83
	Politics	1,392	200	541	651	50.15	22,854	–	–	1.35
	Nature Science	1,193	200	450	543	46.50	14,671	–	–	1.72
	Music	936	100	380	456	48.40	15,441	–	–	1.37
	Literature	916	100	400	416	45.86	11,391	–	–	1.47
	AI	881	100	350	431	39.57	8,260	–	–	1.55
Nested NER	ACE2004	8,512	6,802	813	897	20.12	27,604	12,626	–	2.50
	ACE2005	9,697	7,606	1,002	1,89	17.77	30,711	12,404	–	2.28
	Genia	18,546	15,023	1,669	1,854	25.41	56,015	10,263	–	1.97
Discontinuous NER	CADEC	7,597	5,340	1,097	1,160	16.18	6,316	920	670	2.72
	ShARe13	18,767	8,508	1,250	9,009	14.86	11,148	663	1,088	1.82
	ShARe14	34,614	17,404	1,360	15,850	15.06	19,070	1,058	1,656	1.74

Table 8: Dataset statistics. "#" denotes the amount. "Nes." and "Dis." denote nested and discontinuous entities respectively.

Table 9: Detailed statistics on entity types for thirteen NER datasets.

Datasets	Entity Types
CoNLL2003	location, organization, person, miscellaneous,
OntoNotes	person, norp, facility, organization, gpe, location, product, event, work of art, law, language date, time, percent, money, quantity, ordinal, cardinal
ACE2004	gpe, organization, person, facility, vehicle, location, wea
ACE2005	gpe, organization, person, facility, vehicle, location, wea
Genia	protein, cell_type, cell_line, RNA, DNA,
CADEC	ade
ShARe13	disorder
ShARe14	disorder
Politics	politician, person, organization, political party, event, election, country, location, miscellaneous
Natural Science	scientist, person, university, organization, country, enzyme, protein, chemical compound, chemical element, event, astronomical object, academic journal, award, location, discipline, theory, miscellaneous
Music	music genre, song, band, album, musical artist, musical instrument, award, event, country, location, organization, person, miscellaneous
Literature	writer, award, poem, event, magazine, person, location, book, organization, country, miscellaneous
AI	field, task, product, algorithm, researcher, metrics, university country, person, organization, location, miscellaneous

Datasets / γ	1	5	10	25	50	100
CoNLL2003	93.01	93.26	**93.51**	93.44	93.28	93.16
ACE2005	85.46	**86.41**	86.39	86.30	86.06	85.77
CADEC	70.88	71.34	**71.70**	71.64	71.42	70.99

Table 10: F1 results under different γ using the unified Seq2Seq framework and EnTDA (All).

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section 7

☑ A2. Did you discuss any potential risks of your work?
Section 7

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract, Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 4, Section 5, Appendix A, Appendix B

☑ B1. Did you cite the creators of artifacts you used?
Section 4, Section 5, Appendix A, Appendix B

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Section 4, Section 5, Appendix A, Appendix B

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Section 4, Section 5, Appendix A, Appendix B

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Section 4, Section 5

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section 5.4, Appendix A

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 5.2, Appendix A

C ☑ Did you run computational experiments?

Section 5, Appendix D

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 5.4

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 5.4, Appendix D

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 5.5

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 5.4, Section 5.5, Appendix D

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?
Section 5.5, Appendix F

☑ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Section 5.5, Appendix F

☑ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Section 5.5, Appendix F

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☑ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Section 5.5