Exogenous and Endogenous Data Augmentation for Low-Resource Complex Named Entity Recognition

Xinghua Zhang
Institute of Information Engineering, Chinese Academy of Sciences & School of Cyber Security, University of Chinese Academy of Sciences
Beijing, China
zhangxinghua@iie.ac.cn

Gaode Chen
Institute of Information Engineering, Chinese Academy of Sciences & School of Cyber Security, University of Chinese Academy of Sciences
Beijing, China
chengaode@iie.ac.cn

Shiyao Cui
Institute of Information Engineering, Chinese Academy of Sciences & School of Cyber Security, University of Chinese Academy of Sciences
Beijing, China
cuishiyao@iie.ac.cn

Jiawei Sheng
Institute of Information Engineering, Chinese Academy of Sciences & School of Cyber Security, University of Chinese Academy of Sciences
Beijing, China
shengjiawei@iie.ac.cn

Tingwen Liu*
Institute of Information Engineering, Chinese Academy of Sciences & School of Cyber Security, University of Chinese Academy of Sciences
Beijing, China
liutingwen@iie.ac.cn

Hongbo Xu
Institute of Information Engineering, Chinese Academy of Sciences & School of Cyber Security, University of Chinese Academy of Sciences
Beijing, China
hbxu@iie.ac.cn

ABSTRACT

Low-resource Complex Named Entity Recognition aims to detect entities with the form of any linguistic constituent under scenarios with limited manually annotated data. Existing studies augment the text through the substitution of same type entities or language modeling, but suffer from the lower quality and the limited entity context patterns within low-resource corpora. In this paper, we propose a novel **d**ata **a**ugmentation method **E^{2}DA** from both **e**xogenous and **e**ndogenous perspectives. As for exogenous augmentation, we treat the limited manually annotated data as anchors, and leverage the powerful instruction-following capabilities of Large Language Models (LLMs) to expand the anchors by generating data that are highly dissimilar from the original anchor texts in terms of entity mentions and contexts. As regards the endogenous augmentation, we explore diverse semantic directions in the implicit feature space of the original and expanded anchors for effective data augmentation. Our complementary augmentation method from two perspectives not only continuously expands the global text-level space, but also fully explores the local semantic space for more diverse data augmentation. Extensive experiments on 10 diverse datasets across various low-resource settings demonstrate that the proposed method excels significantly over prior state-of-the-art data augmentation methods.

CCS CONCEPTS

• **Computing methodologies → Information extraction**; • **Information systems → Information retrieval query processing**.

KEYWORDS

Knowledge Acquisition, Data Augmentation, Named Entity Recognition, Low-resource learning

ACM Reference Format:
Xinghua Zhang, Gaode Chen, Shiyao Cui, Jiawei Sheng, Tingwen Liu, and Hongbo Xu. 2024. Exogenous and Endogenous Data Augmentation for Low-Resource Complex Named Entity Recognition. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24), July 14–18, 2024, Washington, DC, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3626772.3657754

1 INTRODUCTION

Named Entity Recognition (NER) is a core task in information extraction, which benefits web search [6, 12, 21, 26, 47], recommendation [18, 50], entity set expansion [30, 32, 48, 49, 52], entity labeling [17, 34] and so on [10, 25, 31, 36, 54]. NER is to detect the entity mention in a text and assign it a predefined category (e.g., *location*, *group*). Complex named entities which are collected from search queries and take the form of any linguistic constituent such as imperative clause (e.g., movie "Dial M for Murder"), pose challenges for NER systems, and have been recently attracting extensive attention [24]. Complex NER plays a vital role in advancing the application of NER tasks, and low-resource settings further extend its applicability to a broader spectrum of scenarios.

Data augmentation technique has been shown to be a promising way for low-resource scenarios [13, 44]. Prior studies explore data augmentation for NER, which can be broadly classified into two types: *substitution based* and *language model based* methods. Substitution based method [8, 43, 46] replaces entities or tokens with other ones of the same type, which are sampled from the training corpus or external knowledge base (e.g., synonyms retrieved from WordNet [11]). Language model based methods generate the augmented data based on discriminative or generative language models. Some of the methods mask and replace entity tokens using the mask language modeling (MLM) of discriminative language

[*] Corresponding author.

SIGIR '24, July 14–18, 2024, Washington, DC, USA
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0431-4/24/07.
https://doi.org/10.1145/3626772.3657754

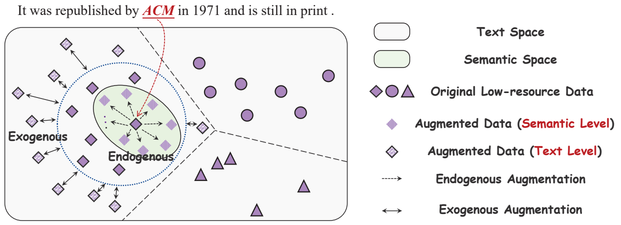

It was republished by ACM in 1971 and is still in print .
Exogenous
Endogenous
Text Space
Semantic Space
Original Low-resource Data
Augmented Data (Semantic Level)
Augmented Data (Text Level)
Endogenous Augmentation
Exogenous Augmentation

Figure 1: Our method E^{2}DA explicitly generates more diverse textual data which are highly dissimilar from original low-resource data (*Exogenous Augmentation*), and searches for diverse implicit semantic directions based on representation of each word in the sentence (*Endogenous Augmentation*).

model [22, 55]. In the other part, generative language models unleash their abilities of next token prediction for producing new sentences [9, 14, 23]. However, existing data augmentation methods for complex NER encounter limitations in two critical aspects: (1) **Limited diversity of augmented data**: the augmented data derive from the low-resource training corpus via entity replacement or language model generation, and then possess the similar entity context patterns with original low-resource data, which leads to the poorer data augmentation diversity. (2) **Limited quantity of augmented data**: current methods can explicitly generate as much text as possible, but may be computationally intensive and suffer from context-entity mismatch issue or incoherent augmentations. Excessive augmented data greatly increase the risk of introducing text noise. This also leads to only a limited amount of augmented data being utilized due to lower data augmentation quality.

To this end, we develop the exogenous and endogenous data augmentation method **E^{2}DA** for low-resource complex NER built upon *vector decomposition based task disentanglement* framework, which decomposes NER task into entity detection and type prediction sub-task for reducing the learning difficulty of complex entities and providing better basic performance especially under low-resource scenarios. The *exogenous data augmentation* and *endogenous data augmentation* processes are as follows:

(1) **Exogenous Data Augmentation**: compared to traditional pre-trained language models such as T5 [29] and BART [20], large language models (LLMs) like ChatGPT have demonstrated the impressive instruction-following ability. Therefore, we design the instruction for prompting LLM to generate new data which are highly dissimilar from the original low-resource training corpus in terms of entity mentions and contexts, and propose the self-reflection strategy for refining the quality of the augmented data. Previous methods whether knowledge base based entity substitution or language model based new text construction for data augmentation, are essentially the simple extension of the original training data where the augmented data have similar entity context patterns. Instead, our LLM based data augmentation, on the one hand, generates more diverse data through explicit instruction constraints, and on the other hand, offers the high usability and scalability, eliminating the need for retraining language models.

(2) **Endogenous Data Augmentation**: prior work have confirmed that there exist many semantic directions in the deep feature space [3, 5, 37]. Moving a data sample along these directions changes its features to match another sample of the same class but with different semantics [16, 39]. For example, the semantic shift of the group entity "ACM" for a certain direction may correspond to another group entity "Springer". Therefore, in two sub-tasks, we can respectively obtain rich semantic features by searching for such semantic directions on task-specific representations and then effectively augment the training data as shown in Figure 1, which reduces the risk of introducing the noisy text. However, it is not a trivial work to look for such semantic directions. To capture the meaningful semantic directions, we estimate the covariance matrix for each class considering both entity mention and context to model the intra-class semantic variations. Following this, the semantic features from different directions are sampled based on a normal distribution with the features of training samples as the mean and the estimated matrix as the covariance. Finally, we derive a upper bound of the loss function to utilize almost infinite semantic features in diverse directions by directly minimizing the upper bound, which contributes to the utilization of more quantity of augmented data at the level of semantic space and unleashes the great potential of data augmentation.

The major contributions of this paper are summarized as follows:

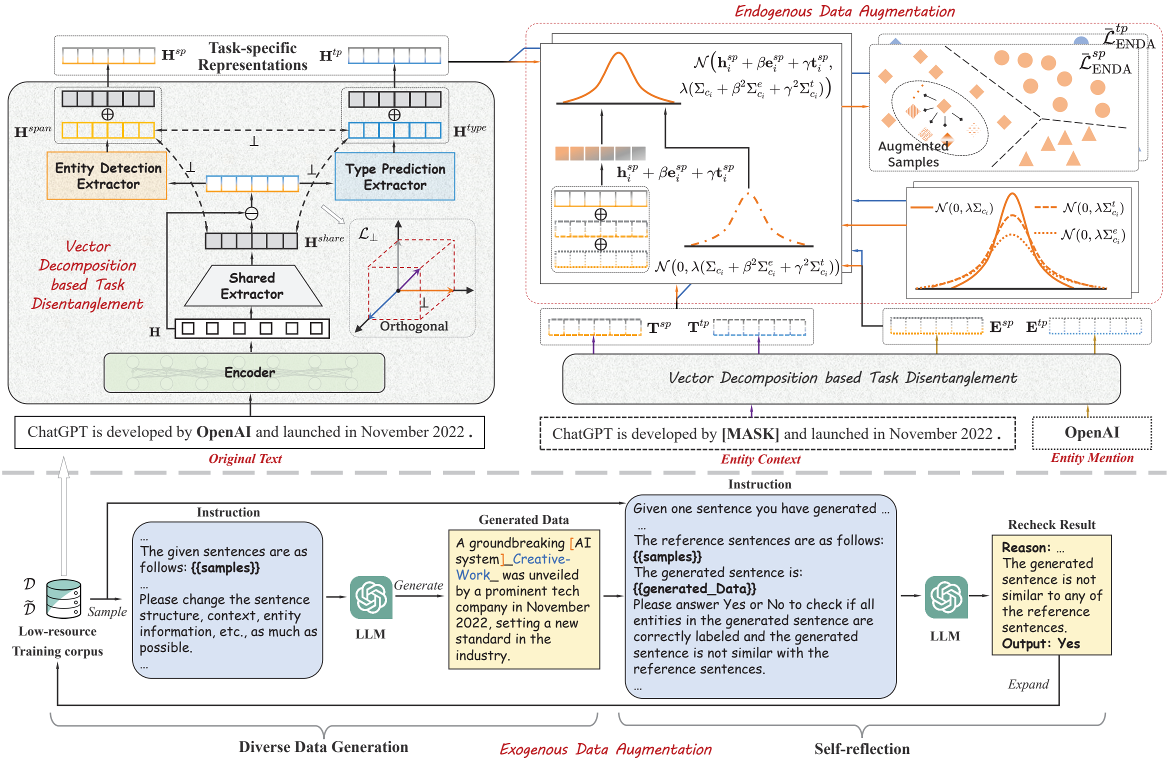
- Unlike prior data augmentation methods which explicitly generate textual content, we first perform data augmentation at the level of semantic space for NER. This augmentation fully explores meaningful semantic directions and utilizes almost infinite semantic features in the implicit feature space by minimizing the upper bound of derived loss function, alleviating the text noise and efficiently employing more comprehensive augmentations.
- We first explore the ability of large language model to conduct data augmentations for complex entities, which generates highly diverse data samples according to the tailor-designed instruction constraint and self-reflection strategy. This exogenous data augmentation spreads the original low-resource data (anchor) to a broader space, significantly improving the diversity.
- We conduct extensive experiments on 10 datasets across four low-resource settings and confirm the significant superiority of our method [1] (average 7.84% absolute increase at most).

2 RELATED WORK

Complex Named Entity Recognition (NER) presents significant challenges, as its context is less informative [2, 15] and its entities are syntactically ambiguous and linguistically complex [1, 14], such as infinitives (e.g., To Kill a Mockingbird). Low-resource scenarios are closer to real applications, but also pose greater difficulties [35, 53]. Data augmentation has emerged as a promising solution to the low-resource NER. Existing data augmentation studies can be grouped into substitution based and language model based methods.

Substitution based Augmentation. These methods explore effective data augmentations by replacing entities (tokens) with existing entities (tokens) of the same type retrieved from the original corpus or external knowledge base (e.g., WordNet [11]). For example, Dai and Adel [8] designed several simple replacement strategies, including synonym and mention replacement, etc. Wu et al. [43] investigated the token substitution and mixup technique with a unified meta-reweighting framework. Xu et al. [46] explored

[1] The code and data are available at https://github.com/AIRobotZhang/E2DA.


Endogenous Data Augmentation
Task-specific Representations
$\mathbf{H}^{sp}$
$\mathbf{H}^{tp}$
$\mathbf{H}^{span}$
$\mathbf{H}^{type}$
$\perp$
Entity Detection Extractor
Type Prediction Extractor
$\mathbf{H}^{share}$
Shared Extractor
$\mathcal{L}_{\perp}$
Orthogonal
Vector Decomposition based Task Disentanglement
$\mathbf{H}$
Encoder
$\mathcal{N}\big(\mathbf{h}_i^{sp} + \beta\mathbf{e}_i^{sp} + \gamma\mathbf{t}_i^{sp}, \lambda(\Sigma_{c_i} + \beta^2\Sigma_{c_i}^{e} + \gamma^2\Sigma_{c_i}^{t})\big)$
$\mathbf{h}_i^{sp} + \beta\mathbf{e}_i^{sp} + \gamma\mathbf{t}_i^{sp}$
$\mathcal{N}(0, \lambda(\Sigma_{c_i} + \beta^2\Sigma_{c_i}^{e} + \gamma^2\Sigma_{c_i}^{t}))$
$\overline{\mathcal{L}}_{\mathrm{ENDA}}^{tp}$
$\overline{\mathcal{L}}_{\mathrm{ENDA}}^{sp}$
Augmented Samples
$\mathcal{N}(0, \lambda\Sigma_{c_i})$
$\mathcal{N}(0, \lambda\Sigma_{c_i}^{t})$
$\mathcal{N}(0, \lambda\Sigma_{c_i}^{e})$
$\mathbf{T}^{sp}$
$\mathbf{T}^{tp}$
$\mathbf{E}^{sp}$
$\mathbf{E}^{tp}$
Vector Decomposition based Task Disentanglement
Original Text
Entity Context
Entity Mention
$\mathcal{D}$
$\tilde{\mathcal{D}}$
Sample
Low-resource Training corpus
Instruction
LLM
Generate
Generated Data
Instruction
LLM
Recheck Result
Expand
Diverse Data Generation
Exogenous Data Augmentation
Self-reflection

ChatGPT is developed by **OpenAI** and launched in November 2022 .

ChatGPT is developed by **[MASK]** and launched in November 2022 .

OpenAI

...
The given sentences are as follows: **{{samples}}**
...
Please change the sentence structure, context, entity information, etc., as much as possible.
...

A groundbreaking [AI system]_Creative-Work_ was unveiled by a prominent tech company in November 2022, setting a new standard in the industry.

Given one sentence you have generated ...
...
The reference sentences are as follows:
{{samples}}
The generated sentence is:
{{generated_Data}}
Please answer Yes or No to check if all entities in the generated sentence are correctly labeled and the generated sentence is not similar with the reference sentences.
...

Reason: ...
The generated sentence is not similar to any of the reference sentences.
Output: Yes

Figure 2: Overview of E^{2}DA, which is comprised of *Vector Decomposition based Task Disentanglement* **for task-specific representations,** *Exogenous Data Augmentation* **for diverse text-level samples, and** *Endogenous Data Augmentation* **for infinite semantic features.**

the peer relation in which two entities are instances of the same category and share similar features, and employed entity pairs with peer relation as the augmentation data for training.

Language Model based Augmentation. Language model based methods produce the augmented data by utilizing the mask language modeling [22], or generating entirely new sentences with generative language modeling [4, 9, 23]. Zhou et al. [55] injected NER labels into the sentence for enabling the mask language modeling to explicitly exploit label information, and then synthesized the augmented data with novel entities. Ghosh et al. [14] masked all other words in the sentence except the entities and keywords for a new text reconstruction to generate diverse-pattern sentences by combining existing text patterns. Recently, there has been a growing trend of utilizing large language models (LLMs) for data augmentation [28, 38, 40]. However, the effect of data augmentation remains unexplored for complex NER task.

Different from prior data augmentation methods that suffer from high complexity of complex entities, and then generate low-quality and poor-diversity data, we propose a novel augmentation method E^{2}DA, which for the first time explores the meaningful semantic directions for NER by minimizing a high-efficient loss upper bound, and evokes the ability of LLMs to augment data with our tailor-designed instructions and self-reflection strategies. We unleash the complementarity of local semantic features and global text-level data augmentations, increasing the efficacy of data augmentations and inspiring a new perspective on NER data augmentation.

3 METHODOLOGY

We formally describes the low-resource complex named entity recognition task in Sec. 3.1, and then depict our exogenous and endogenous data augmentation method **E^{2}DA**, which continuously expands the original low-resource training data (anchor) and then explores the vast seman tic space surrounding each sample. In Figure 2, we give the overview of **E^{2}DA**, where ***Vector Decomposition based Task Disentanglement*** provides the task-specific representations for each sample, ***Exogenous Augmentation*** generates diverse textual data relying on powerful instruction-following ability of large language models (LLMs) and ***Endogenous Augmentation*** explores the broad semantic space across meaningful semantic directions for utilizing almost infinite semantic features.

3.1 Task Description

Named Entity Recognition (NER) aims to detect the entity e with the corresponding category from a sentence S=<$s_1, s_2, ..., s_n$>. s_i is a word (token) and n is the length of S. An entity e is a text mention in S with an entity category c (e.g., *location*, *group*): $e =< (s_{start}, s_{start+1}, ..., s_{end}), c >$, where $c \in C^{tp}$, and C^{tp} is the predefined entity category set. The entity boundary is labeled by choosing from C^{sp} = {B, I, O} where B indicates the first token of an entity, I marks the other part of the entity, and O indicates the non-entity token. A sentence S may have one or more entities, or none at all.

Complex named entities, such as titles of creative works (e.g., movie and book names), are more than just simple nouns, making their recognition notably challenging [1]. They can take the form of any linguistic constituent, like an imperative clause ("Dial M for Murder"), and do not look like traditional named entities (e.g., person names, location). This syntactic ambiguity makes it challenging to recognize them based on their context [24]. Low-resource complex named entity recognition involves learning a model with only a limited amount (K) of labeled training data $\mathcal{D}$ available (e.g., K = 100, 200), further posing rigorous challenges for entity recognition.

3.2 Vector Decomposition based Task Disentanglement

Complex NER is a difficult task that is reflected in both less rich context and complicated entities. Complex NER benchmark datasets are curated from search queries or voice commands, the context is less informative and lacks surface features, and complex entities like movie names are syntactically ambiguous and linguistically complex, such as infinitives (e.g., To Kill a Mockingbird). In low-resource situations, the challenge intensifies. Therefore, we first propose a basic entity recognizer for reducing the difficulty of task learning via vector decomposition based task disentanglement, which leads to the smaller label space and purer task modeling.

Basic Encoder. Firstly, the sentence S is encoded by pre-trained language model. Specifically, $S = <s_1, s_2, ..., s_n>$ is input into encoder to extract the contextual hidden representations of all words $\mathbf{H} = <\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_n> \in \mathbb{R}^{n \times d}$ as:

$$\mathbf{H} = \text{Encoder}(S) \tag{1}$$

where d is the dimension of the last hidden layer.

Vector Decomposition. Task disentanglement aims to divide the NER task into *entity detection* and *type prediction* sub-task, which needs task-specific representations for separate sub-task modeling. *Entity detection* sub-task detects the entity span (boundary) and *type prediction* sub-task determines the entity type (category) for each entity span. The vector decomposition, which breaks one vector into two or more components (e.g., $\overrightarrow{A} = \overrightarrow{B} + \overrightarrow{C} + \overrightarrow{D}$), is similar in spirit to disentanglement [42, 45, 51]. Thus, we exploit the vector disentanglement for obtaining task-specific representations. That is the contextual hidden representation $\mathbf{h}_i$ can be viewed as the combination of entity span, type and shared features ($\mathbf{h}_i = \mathbf{h}_i^{span} + \mathbf{h}_i^{type} + \mathbf{h}_i^{share}$). $\mathbf{h}_i^{share}$ is extracted from $\mathbf{h}_i$ based a shared extractor, and then we can obtain the combination of two pure sub-task features $\mathbf{h}_i^{st}$ by the differences between the initial contextual representations $\mathbf{h}_i$ and task-shared representations $\mathbf{h}_i^{share}$ as follows:

$$\mathbf{H}^{share} = \text{Extractor}_{shared}(\mathbf{H}), \quad \mathbf{h}_i^{st} = \mathbf{h}_i - \mathbf{h}_i^{share} \tag{2}$$

where $\mathbf{H}^{share} = <\mathbf{h}_1^{share}, \mathbf{h}_2^{share}, ..., \mathbf{h}_n^{share}>$.

We then utilize the other two extractors to respectively get pure entity span and type features based on the combination of two pure sub-task features $\mathbf{H}^{st} = <\mathbf{h}_1^{st}, \mathbf{h}_2^{st}, ..., \mathbf{h}_n^{st}>$:

$$\mathbf{H}^{span} = \text{Extractor}_{span}(\mathbf{H}^{st}), \quad \mathbf{H}^{type} = \text{Extractor}_{type}(\mathbf{H}^{st}) \tag{3}$$

where $\mathbf{H}^{span} = <\mathbf{h}_1^{span}, \mathbf{h}_2^{span}, ..., \mathbf{h}_n^{span}>$, $\mathbf{H}^{type} = <\mathbf{h}_1^{type}, \mathbf{h}_2^{type}, ..., \mathbf{h}_n^{type}>$. Finally, the two task-specific representations can be respectively obtain via combining the shared features and respective pure task features:

$$\mathbf{H}^{sp} = \mathbf{H}^{span} + \mathbf{H}^{share}, \quad \mathbf{H}^{tp} = \mathbf{H}^{type} + \mathbf{H}^{share} \tag{4}$$

where $\mathbf{H}^{sp} = <\mathbf{h}_1^{sp}, \mathbf{h}_2^{sp}, ..., \mathbf{h}_n^{sp}>$, $\mathbf{H}^{tp} = <\mathbf{h}_1^{tp}, \mathbf{h}_2^{tp}, ..., \mathbf{h}_n^{tp}>$. To ensure the orthogonality between vectors, we minimize the square of dot product of pairwise vectors among the two pure sub-task representations $\mathbf{h}_i^{span}$, $\mathbf{h}_i^{type}$ and task-shared representations $\mathbf{h}_i^{share}$:

$$\mathcal{L}_{\perp} = \left\|\mathbf{h}_i^{span} \cdot \mathbf{h}_i^{share}\right\|^2 + \left\|\mathbf{h}_i^{type} \cdot \mathbf{h}_i^{share}\right\|^2 + \left\|\mathbf{h}_i^{span} \cdot \mathbf{h}_i^{type}\right\|^2 \tag{5}$$

3.3 Exogenous Data Augmentation

Different from substitution of entities of the same type and new text construction based on traditional pre-trained language models, off-the-shelf large language models (LLMs) can generate more fluent and diverse data with tailor-designed instructions, alleviating the low quality and increasing the diversity of the augmented data without extra retraining. Therefore, we propose the exogenous data augmentation method based on LLMs.

Diverse Data Generation. First, we draw a sample of a observations as the reference sentences from the low-resource training data $\mathcal{D}$. Given a prompt π_0 and template $\mathsf{F}(\cdot)$, we can get the instruction $\mathsf{F}(samples, \pi_0)$ and query the LLM to generate the new data:

$$\widetilde{S} \leftarrow \text{LLM}(\mathsf{F}(samples, \pi_0)) \tag{6}$$

where $\mathsf{F}(\cdot)$ aims to fill the sampled data into the slots of prompt π_0. This process can generate multiple data by sampling and querying the LLM multiple times. By adding the explicit descriptions into prompt π_0 for requesting to generate highly dissimilar data from the original corpus, such as "*Please change the sentence structure, context, entity information, etc., as much as possible*", we can initially obtain the diverse augmented data $\widetilde{S}$. For the format of the generated data, we can provide clear format definitions in prompt π_0 for instructing LLM to output in the pre-defined format. As shown in Figure 2, the output format is that the entity is marked with "[", "]" and followed with its entity type. According to the pre-defined format, we can parse the corresponding entities and their labels from the generated data, thereby obtaining the augmented data with labels.

Self-Reflection Strategy. Recent studies have shown that LLM possesses inherent reflective capacities to refine knowledge to a certain degree [19, 33]. Inspired by this, we develop the self-reflection strategy to recheck the generated data of LLM, such as the correctness of labels and the diversity of text, to further improve the quality of the generated data. Similarly, given the prompt π_1 and template $\mathsf{F}(\cdot)$, we can get the instruction $\mathsf{F}(samples, \widetilde{S}, \pi_1)$, and then query the LLM to recheck if $\widetilde{S}$ is correctly labeled and dissimilar from original low-resource data by giving the reason and final decision:

$$\begin{aligned} R &\leftarrow \text{LLM}(\mathsf{F}(samples, \widetilde{S}, \pi_1)) \\ \widetilde{\mathcal{D}} &\leftarrow \begin{cases} \widetilde{\mathcal{D}} \cup \widetilde{S}, & \text{if } R \text{ is "Yes"} \\ \widetilde{\mathcal{D}}, & \text{otherwise} \end{cases} \end{aligned} \tag{7}$$

where $R \in$ {"Yes", "No"} and $\widetilde{S}$ will be expanded into the exogenous augmentation corpus $\widetilde{\mathcal{D}}$ if R is "Yes".

3.4 Endogenous Data Augmentation

Previous methods explicitly generate multiple samples by manipulating the data augmentor many times. Nevertheless, the amount of samples it generates is still limited and the generated data also suffer from context-entity mismatch issue or incoherent augmentations especially in complex NER. Instead, we propose endogenous data augmentation from the semantic space perspective, which does not need to explicitly generate textual content but explores more meaningful and diverse semantic directions based on the task-specific representations $\mathbf{H}^{sp}$ or $\mathbf{H}^{tp}$. Next, we describe the endogenous data augmentation procedure which is adopted in the same way in two sub-tasks (*entity detection* and *type prediction*), and use $\mathbf{H}^{\mathsf{r}}$ to refer to the task-specific representations, use C to refer to C^{sp} or C^{tp}.

Intra-class Semantic Variations. Concretely, we first estimate a covariance matrix Σ_{c_i} for each class c_i based on the features of all the samples in class c_i. Then, the augmented representation $\widetilde{\mathbf{h}}_i^{\mathsf{r}}$ for each token s_i is obtained by turning $\mathbf{h}_i^{\mathsf{r}}$ along a random direction sampled from a normal distribution $\mathcal{N}(0, \lambda\Sigma_{c_i})$ as:

$$\widetilde{\mathbf{h}}_i^{\mathsf{r}} \sim \mathcal{N}(\mathbf{h}_i^{\mathsf{r}}, \lambda\Sigma_{c_i}) \tag{8}$$

where λ is the hyper-parameter and controls the strength of data augmentation ($\lambda > 0$). Next, a straightforward way to generate as much augmentations as possible is to sample $\mathtt{m}$ times from the distribution in Eq. 8. Then, we can train the entity recognition model with the standard cross-entropy loss as follows:

$$\mathcal{L}_{\mathtt{m}} = -\frac{1}{n}\sum_{i=1}^{n}\frac{1}{\mathtt{m}}\sum_{k=1}^{\mathtt{m}} log\frac{\exp(\mathbf{w}_{c_i}^{\top}\mathbf{h}_{i,k}^{\mathsf{r}} + b_{c_i})}{\sum_{c_j \in C}\exp(\mathbf{w}_{c_j}^{\top}\mathbf{h}_{i,k}^{\mathsf{r}} + b_{c_j})} \tag{9}$$

where $[\mathbf{w}_{c_i}^{\top}; b_{c_i}]$ are parameters of classification head specific to the class c_i in the sub-task. It is worth noting that two sub-tasks have different classification heads. $\mathbf{h}_{i,k}^{\mathsf{r}}$ indicates the k-th sampling based on Eq. 8 for $\mathbf{h}_i^{\mathsf{r}}$. However, above procedure still needs to explicitly augment multiple times, and the sampling variance is unstable and limited for low-resource data when $\mathtt{m}$ is a finite number.

Utilization of Almost Infinite Semantic Features. Therefore, we set $\mathtt{m} \longrightarrow \infty$ for considering all meaningful features across diverse semantic directions, the loss can be formalized as:

$$\mathcal{L}_{\text{ENDA}} = -\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_{\widetilde{\mathbf{h}}_i^{\mathsf{r}}}\left[log\frac{\exp(\mathbf{w}_{c_i}^{\top}\widetilde{\mathbf{h}}_i^{\mathsf{r}} + b_{c_i})}{\sum_{c_j \in C}\exp(\mathbf{w}_{c_j}^{\top}\widetilde{\mathbf{h}}_i^{\mathsf{r}} + b_{c_j})}\right] \tag{10}$$

However, Eq. 10 takes the mathematical expectations and is difficult to implement and compute. Therefore, we make the following derivations to get the upper bound of Eq. 10:

(1) Following the operation rules for exponential and logarithmic functions, the equation can be directly transformed as follows:

$$\mathcal{L}_{\text{ENDA}} = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_{\widetilde{\mathbf{h}}_i^{\mathsf{r}}}\left[log\sum_{c_j \in C}\exp((\mathbf{w}_{c_j}^{\top} - \mathbf{w}_{c_i}^{\top})\widetilde{\mathbf{h}}_i^{\mathsf{r}} + (b_{c_j} - b_{c_i}))\right] \tag{11}$$

(2) According to Jensen's inequality $\mathbb{E}\left[logX\right] \leq log\mathbb{E}\left[X\right]$, we can further derive as:

$$\mathcal{L}_{\text{ENDA}} \leq \frac{1}{n}\sum_{i=1}^{n}log\sum_{c_j \in C}\mathbb{E}_{\widetilde{\mathbf{h}}_i^{\mathsf{r}}}\left[\exp((\mathbf{w}_{c_j}^{\top} - \mathbf{w}_{c_i}^{\top})\widetilde{\mathbf{h}}_i^{\mathsf{r}} + (b_{c_j} - b_{c_i}))\right] \tag{12}$$

(3) According to the Moment Generating Function (MGF) [27], that is $M_X(t) = \mathbb{E}\left[\exp(tX)\right] = \exp(t\mu + \frac{1}{2}\sigma^2 t^2), X \sim \mathcal{N}(\mu, \sigma^2)$. Exactly seen from Eq. 8 and 12, $\underbrace{(\mathbf{w}_{c_j}^{\top} - \mathbf{w}_{c_i}^{\top})\widetilde{\mathbf{h}}_i^{\mathsf{r}} + (b_{c_j} - b_{c_i})}_{X} \sim$ $\mathcal{N}(\underbrace{(\mathbf{w}_{c_j}^{\top} - \mathbf{w}_{c_i}^{\top})\mathbf{h}_i^{\mathsf{r}} + (b_{c_j} - b_{c_i})}_{\mu}, \underbrace{\lambda(\mathbf{w}_{c_j}^{\top} - \mathbf{w}_{c_i}^{\top})\Sigma_{c_i}(\mathbf{w}_{c_j} - \mathbf{w}_{c_i})}_{\sigma^2})$, and $t = 1$. Furthermore, Eq. 12 can be written as follows:

$$\begin{aligned}\mathcal{L}_{\text{ENDA}} \leq \bar{\mathcal{L}}_{\text{ENDA}} = \frac{1}{n}\sum_{i=1}^{n}log\sum_{c_j \in C}\exp((\mathbf{w}_{c_j}^{\top} - \mathbf{w}_{c_i}^{\top})\mathbf{h}_i^{\mathsf{r}} \\ +(b_{c_j} - b_{c_i}) + \frac{\lambda}{2}(\mathbf{w}_{c_j}^{\top} - \mathbf{w}_{c_i}^{\top})\Sigma_{c_i}(\mathbf{w}_{c_j} - \mathbf{w}_{c_i}))\end{aligned} \tag{13}$$

(4) Under low-resource scenarios, the amount of data for each class is limited which may impair the estimate of covariance matrix. Therefore, we further explicitly take entity mention and its context into account for capturing more precise intra-class semantic variations. For entity mention itself, we only input the entity into the *vector decomposition based task disentanglement* framework for task-specific representations $\mathbf{E}^{\mathsf{r}} = <\mathbf{e}_1^{\mathsf{r}}, \mathbf{e}_2^{\mathsf{r}}, ..., \mathbf{e}_l^{\mathsf{r}}>$. For its context, we mask the entity mention and input it into the same framework for task-specific representations $\mathbf{T}^{\mathsf{r}} = <\mathbf{t}_1^{\mathsf{r}}, \mathbf{t}_2^{\mathsf{r}}, ..., \mathbf{t}_n^{\mathsf{r}}>$. Then we can respectively get the inequality in Eq. 12 for entity mention and context. According to the property of multiple random variables in the Moment Generating Function (MGF): $Z = X + \beta Y$, $M_Z(t) = M_X(t) \cdot M_Y(\beta t)$, we can further strengthen the upper bound of endogenous augmentations with entity mention and context for each entity token as follows:

$$\begin{aligned}\bar{\mathcal{L}}_{\text{ENDA}} = \frac{1}{n}\sum_{i=1}^{n}log\sum_{c_j \in C}\exp((\mathbf{w}_{c_j}^{\top} - \mathbf{w}_{c_i}^{\top})\mathbf{z}_i + \eta(b_{c_j} - b_{c_i}) \\ +\frac{\lambda}{2}(\mathbf{w}_{c_j}^{\top} - \mathbf{w}_{c_i}^{\top})\Psi_{c_i}(\mathbf{w}_{c_j} - \mathbf{w}_{c_i}))\end{aligned} \tag{14}$$

where $\mathbf{z}_i = \mathbf{h}_i^{\mathsf{r}} + \beta\mathbf{e}_i^{\mathsf{r}} + \gamma\mathbf{t}_i^{\mathsf{r}}, \eta = 1 + \beta + \gamma, \Psi_{c_i} = \Sigma_{c_i} + \beta^2\Sigma_{c_i}^{e} + \gamma^2\Sigma_{c_i}^{t}$. $\Sigma_{c_i}^{e}$ is the estimated covariance matrix only based on entity features $\mathbf{e}_i^{\mathsf{r}}$ of all the samples in class c_i, and $\Sigma_{c_i}^{t}$ is estimated by context features $\mathbf{t}_i^{\mathsf{r}}$. β and γ are hyper-parameters which control the strength of semantic augmentations for entity mention and context.

3.5 Training Optimization and Prediction

This section describes how to optimize our method E^{2}DA during training and how to infer during prediction phase.

Training Optimization. The overall training objective, which is minimized during training, is defined as follows:

$$\mathcal{L} = \bar{\mathcal{L}}_{\text{ENDA}}^{sp} + \bar{\mathcal{L}}_{\text{ENDA}}^{tp} + \alpha\mathcal{L}_{\perp} \tag{15}$$

where $\bar{\mathcal{L}}_{\text{ENDA}}^{sp}$ and $\bar{\mathcal{L}}_{\text{ENDA}}^{tp}$ are respectively the upper bound of endogenous augmentation for *entity detection* and *type prediction* sub-tasks based on Eq. 14. α is the hyper-parameter. Considering the data quality generated by the LLM, we first train the model on the exogenous augmentation corpus $\widetilde{\mathcal{D}}$ and then fine-tune it on the original low-resource training data $\mathcal{D}$, both based on Eq. 15.

Table 1: The statistics of datasets with 4 low-resource settings.

	En	Bn	Hi	De	Es
#Train	K=100, 200, 500, 1000				
#Dev	800	800	800	800	800
#Test	217,818	133,119	141,565	217,824	217,887
	Ko	Nl	Ru	Tr	Zh
#Train	K=100, 200, 500, 1000				
#Dev	800	800	800	800	800
#Test	178,249	217,337	217,501	136,935	151,661

Prediction. We use the *Softmax* function to get probability distributions for each word (token) s_i in two sub-tasks respectively:

$$p(c|s_i) = \frac{\exp\{\mathbf{w}_c^\top \mathbf{h}_i^r + b_{c_i}\}}{\sum_{c_j \in C} \exp\{\mathbf{w}_{c_j}^\top \mathbf{h}_i^r + b_{c_j}\}} \qquad (16)$$

$$\tilde{y} = \arg\max_c p(c|s_i)$$

where $\mathbf{h}_i^r$ is respectively $\mathbf{h}_i^{sp}$ and $\mathbf{h}_i^{tp}$, and C is respectively C^{sp} = {B, I, O} and C^{tp} = {*location*, *group*, ..., O} in entity detection and type prediction sub-tasks. $\tilde{y}$ is the predicted label in two sub-tasks. Therefore, during prediction phase, we can get the entity mentions for each sentence $S =< s_1, s_2, ..., s_n >$ based on the predicted labels in entity detection sub-task. Based on the detected entities, we use the predicted label of the rightmost token for each entity in type prediction sub-task, which is viewed as the entity type of the whole entity. If the predicted entity type is O, the entity is discarded.

4 EXPERIMENTS

We aim to answer the following research questions as follows:

- **RQ1**: Has E^{2}DA shown a notable improvement in performance?
- **RQ2**: How significant a role have *exogenous augmentations* played?
- **RQ3**: What are the advantages of *endogenous augmentations*?

4.1 Experimental Datasets

We utilize the large multilingual benchmark MultiCoNER [24] for complex NER, which is collected from Bing search queries, questions and Wiki sentences. The dataset represents great challenges in NER due to the complex entity mentions and contexts. Following Ghosh et al. [14], we conduct experiments on a set of 10 languages including English (En), Bengali (Bn), Hindi (Hi), German (De), Spanish (Es), Korean (Ko), Dutch (Nl), Russian (Ru), Turkish (Tr) and Chinese (Zh). Each language dataset has 6 entity types: *person*, *location*, *corporation*, *groups* (such as political party names), *product* (consumer products such as apple iPhone 6), and *creative work* (movie/song/book titles such as Mr. Smith Goes to Washington). We perform the low-resource complex NER experiments on four settings (K=100, 200, 500, 1000) following Ghosh et al. [14], that is the number of training data is respectively 100, 200, 500 and 1000. The validation (Dev) and test dataset are the same under different low-resource settings. The detailed statistics of datasets are shown in Table 1. It is worth noting that the number of data in test sets is large (0.1 million level), which can better testify the effectiveness and generalization of the methods.

4.2 Experimental Settings

4.2.1 Setup and Evaluation. All hyper-parameters are tuned according to the results on dev set with grid-search. The maximum training epoch is 500 and the learning rate is 1e-5. The batch size is set to 16 by tuning from {8, 16, 32}. α is set to 0.01, β and γ are all set to 0.1 by tuning from {0.05, 0.1, 0.5, 1.0}. $\lambda = t/T \times \lambda_0$ dynamically increases with the training process due to the inaccurate estimate of covariance in the early stage, where λ_0 is set to 1.5, t is the current epoch and T is the total epochs. The number of the reference sentences a is set to 10, balancing the performance and LLM's limitations (such as input length, and cost). Under different low-resource settings, the amount of augmented data is less than or equal to five times the size of the original training set for fair comparison with the competitive baseline ACLM [14]. Following previous competitive baselines, we use XLM-RoBERTa-large [7] as encoders. Extractor$_{shared}$, Extractor$_{span}$, and Extractor$_{type}$ are single-layer fully connected networks in our E^{2}DA method. LLM is `gpt-3.5-turbo-0613` which is used by querying OpenAI API in our experiments. We implement our method with Pytorch based on huggingface Transformers [41], which is conducted on NVIDIA Tesla V100 GPU. The baseline results are all reported by Ghosh et al. [14]. In line with Ghosh et al. [14], we use the micro-F1 score as the evaluation metric based on exact entity matching.

4.2.2 Baselines. We make comparisons with the following competitive NER data augmentation baselines to confirm the effectiveness of our proposed augmentation method: (1) **Gold-Only** trains the NER model only on the low-resource training data without any data augmentations. (2) **LwTR** [8] uses a label-wise token distribution which is built from the original training set, to randomly select another token with the same label for replacement, and whether to replace is determined by a binomial distribution. (3) **DAGA** [9] first linearizes the labeled sentences, then a language model is trained on the linearized data and used to generate synthetic labeled data. (4) **MELM** [55] first performs labeled sequence linearization to insert the entity label tokens into the NER training sentences, which is used for fine-tuning the masked entity language modeling. Finally, they get the augmented data by generating diverse entities via masked entity prediction. (5) **ACLM** [14] formulates the data augmentation as a conditional generation task where a conditional text generation model generates the augmented data by introducing new and diverse context patterns around an entity based on the original low-resource training data.

4.3 Experimental Results

4.3.1 Main Results (**RQ1**). Table 2 gives the F1 scores on 10 datasets (languages) under four different low-resource settings (100, 200, 500, and 1000). We show the absolute increase of our E^{2}DA compared to ACLM [14] in blue. We also report the average F1 score for each low-resource setting in the last column of Table 2. We can see that our data augmentation method E^{2}DA achieves the notable improvements under four low-resource scenarios.

Overall, the performance of nearly all methods tends to progressively improve as the quantity of training data increases from 100 to 1000 in Table 2. Our E^{2}DA method has more obvious advantages (the improvement of 11.98% at most) under the lower-resource settings (e.g., 100) due to the sufficient augmentations based on almost

Table 2: F1 scores on 10 datasets under four different low-resource settings where the training set respectively contains 100, 200, 500 and 1000 sentences. We mark with *Improv.* and show the absolute increase compared to the previous state-of-the-art ACLM.

#Train	Method	En	Bn	Hi	De	Es	Ko	Nl	Ru	Tr	Zh	Average
100	Gold-only	29.36	14.49	18.80	37.04	36.30	12.76	38.78	23.89	24.13	14.18	24.97
	LwTR [8]	48.60	20.25	29.95	48.38	44.08	35.09	43.00	39.22	30.58	27.70	36.68
	DAGA [9]	16.24	5.87	10.40	32.44	27.78	19.28	15.44	11.14	16.17	10.33	16.51
	MELM [55]	40.12	6.22	27.84	43.94	37.45	34.10	37.82	32.38	20.13	25.11	30.51
	ACLM [14]	48.76	23.09	33.53	48.80	44.14	38.35	46.22	39.48	37.20	35.12	39.47
	E^{2}DA*Improv.*	56.69$_{\uparrow 7.93}$	35.07$_{\uparrow 11.98}$	44.32$_{\uparrow 10.79}$	56.02$_{\uparrow 7.22}$	52.83$_{\uparrow 8.69}$	47.77$_{\uparrow 9.42}$	53.24$_{\uparrow 7.02}$	44.36$_{\uparrow 4.88}$	40.57$_{\uparrow 3.37}$	42.26$_{\uparrow 7.14}$	**47.31**$_{\uparrow 7.84}$
200	Gold-only	51.83	19.31	33.68	49.62	45.16	42.51	47.83	31.55	26.76	32.34	38.06
	LwTR [8]	52.88	23.85	34.27	50.31	47.01	42.77	52.01	40.18	35.92	30.57	40.98
	DAGA [9]	33.30	17.12	19.58	35.10	33.56	26.50	38.04	29.83	23.35	25.66	28.20
	MELM [55]	47.83	5.47	29.67	45.85	42.08	36.62	49.47	41.84	31.25	32.27	36.24
	ACLM [14]	54.99	38.39	40.55	53.36	49.57	44.32	53.19	43.97	39.71	39.31	45.74
	E^{2}DA*Improv.*	58.17$_{\uparrow 3.18}$	40.15$_{\uparrow 1.76}$	43.26$_{\uparrow 2.71}$	57.73$_{\uparrow 4.37}$	56.44$_{\uparrow 6.87}$	48.14$_{\uparrow 3.82}$	57.74$_{\uparrow 4.55}$	48.98$_{\uparrow 5.01}$	43.66$_{\uparrow 3.95}$	40.54$_{\uparrow 1.23}$	**49.48**$_{\uparrow 3.74}$
500	Gold-only	55.51	34.60	38.66	55.95	51.52	48.57	50.97	45.14	38.83	38.84	45.86
	LwTR [8]	56.97	35.42	37.83	55.91	54.74	49.36	56.10	46.82	39.00	38.55	47.07
	DAGA [9]	44.62	22.36	24.30	43.02	42.77	36.23	47.11	30.94	30.84	33.79	35.60
	MELM [55]	52.57	9.46	31.57	53.57	46.40	45.01	51.90	46.73	38.26	39.64	41.51
	ACLM [14]	58.31	40.26	41.48	59.35	55.69	51.56	56.31	49.40	43.57	41.23	49.72
	E^{2}DA*Improv.*	61.94$_{\uparrow 3.63}$	41.47$_{\uparrow 1.21}$	43.65$_{\uparrow 2.17}$	63.26$_{\uparrow 3.91}$	58.67$_{\uparrow 2.98}$	53.65$_{\uparrow 2.09}$	61.11$_{\uparrow 4.80}$	53.70$_{\uparrow 4.30}$	46.34$_{\uparrow 2.77}$	46.00$_{\uparrow 4.77}$	**52.98**$_{\uparrow 3.26}$
1000	Gold-only	57.22	30.20	39.55	60.18	55.86	53.39	60.91	49.93	43.67	43.05	49.40
	LwTR [8]	59.10	39.65	43.90	61.28	57.29	51.37	59.25	52.04	44.33	43.71	51.19
	DAGA [9]	50.24	32.09	35.02	51.45	49.47	42.41	51.88	41.56	33.18	39.51	42.68
	MELM [55]	53.48	6.88	37.02	58.69	52.43	50.50	56.25	48.99	36.83	38.88	44.00
	ACLM [14]	60.14	42.42	48.20	63.80	58.33	55.55	61.22	54.31	48.23	45.19	53.74
	E^{2}DA*Improv.*	62.22$_{\uparrow 2.08}$	45.16$_{\uparrow 2.74}$	50.51$_{\uparrow 2.31}$	66.67$_{\uparrow 2.87}$	60.42$_{\uparrow 2.09}$	56.67$_{\uparrow 1.12}$	64.16$_{\uparrow 2.94}$	56.37$_{\uparrow 2.06}$	48.75$_{\uparrow 0.52}$	52.36$_{\uparrow 7.17}$	**56.33**$_{\uparrow 2.59}$

infinite semantic features and more diversified generated data in the task disentanglement framework. DAGA [9] achieves the poor performance because it may introduce too much text noise (e.g., incoherence). As DAGA [9] trains a LSTM-based recurrent neural network language model (RNNLM) on the low-resource data for generating new sentences, a small amount of data makes it difficult to train language models well, thus generating noisy augmentations, which shows the risk faced by generative language model based methods. Compared to both substitution based (such as LwTR [8]) and language model based methods (e.g., MELM [55], ACLM [14]), our method has achieved significant improvements. The reason may be that those baselines explicitly generate multiple synthesized data to produce the effect, but increase the risk of introducing text noise. And they directly utilize or synthesize the entity context patterns based on the low-resource corpus, which has lower diversity for the augmented data. In comparison with methods without data augmentation (e.g., Gold-only in Table 2), E^{2}DA respectively achieves 22.34%, 11.42%, 7.12%, and 6.93% improvements under 4 low-resource settings, which confirms the necessity of data augmentation and effectiveness of our method. As for training efficiency, the number of processed batches per second (B/s) is 2.2 for E^{2}DA and 3.1 for standard NER model in ACLM [14]. Their prediction efficiency is respectively 19.6 B/s (E^{2}DA) and 21.7 B/s (ACLM).

4.3.2 Ablation Studies. To evaluate the effectiveness of each module in our method, we perform the ablation studies in Table 3. We can see that: (1) Without the constraint of orthogonality (*w/o* $\mathcal{L}_\perp$) among the two task-specific representations and task-shared representations, the performance drops by 1.31%, 1.18%, 1.08%, and 1.14% because the variant can not get more accurate task-specific features and makes the representations vague without the explicit regularization. (2) Vector decomposition based task disentanglement contributes to 1.97%, 2.20%, 1.63%, and 2.28% increases because task decomposition holds the lower learning difficulty in two sub-tasks for complex NER. (3) If we do not consider the entity mention and context for intra-class semantic variations, the performance obviously decreases because low-resource data can not guarantee the precise estimate of covariance, and the exploration of semantic

Table 3: Ablation studies on dev set. The F1 score are averaged over 10 datasets (languages) for each low-resource setting.

Method	Dev F1			
	100	200	500	1000
E^{2}DA (Ours)	**63.64**	**68.01**	**73.06**	**76.33**
w/o $\mathcal{L}_\perp$	62.33	66.83	71.98	75.19
w/o Task Disentanglement	61.67	65.81	71.43	74.05
w/o Entity/Context Enhanced	61.99	66.36	72.56	75.55
w/o Endogenous DA	61.74	66.21	72.15	75.12
w/o Self-Reflection	62.55	66.64	71.36	75.19
w/o Exogenous DA	61.62	66.41	72.66	76.07

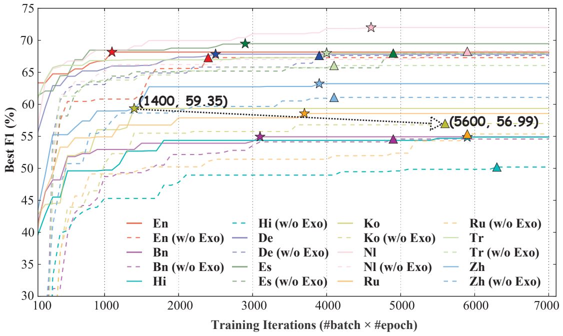


Figure 3: The best performance changes as the training process goes (K=100) with/without *Exo*genous Augmentation.

directions for entity mention and context strengthens the variance estimate. (4) If we replace the endogenous loss $\bar{\mathcal{L}}_{\text{ENDA}}$ with the standard cross-entropy loss (w/o Endogenous DA), F1 scores drop by 1.90%, 1.80%, 0.91%, and 1.21% due to the lack of utilizing infinite semantic features. (5) The self-reflection strategy contributes to an average gain of 1.33% F1 score, resulting from the rechecking ability of the LLM by the tailor-designed instructions which selects the more valuable data from the generated data by the LLM. (6) Exogenous data augmentation generates diverse data and then contributes to the reliable improvements. By comparing the last two rows, we observe that low-quality augmentations are more likely to have a negative impact under the higher-resource settings.

4.4 Experimental Analyses

4.4.1 Impact of Exogenous Augmentation during Training Process (**RQ2**). In Figure 3, we show the best F1 scores on dev sets with 100 training data as the training process goes. Our E^{2}DA with/without exogenous augmentation are respectively described with solid and dashed lines. We also mark the best F1 score throughout the entire training process with ★/△ for each dataset where the results on the same dataset are marked in the identical color. We can observe that the exogenous augmentation speeds the convergence of the model where it achieves higher performance in the early stages and attains optimal performance at the fastest speed with the support of the exogenous augmentations. For example, E^{2}DA achieves the best F1 score of 59.35% in the 1400-*th* iteration, while it only reaches 56.99% with longer iteration rounds (5600 iterations) without exogenous augmentation in Korean (Ko). This indicates that the exogenous augmentation generates higher-quality and more diverse data, which helps the model learn the task more comprehensively.

4.4.2 Impact of Endogenous Augmentation during Training Process (**RQ3**). In Figure 4, we depict the learning curves of F1 scores on dev sets during training when the number of training samples is 100. We show the learning curves of our basic model with or without endogenous augmentations, which are respectively marked with solid and dashed lines for each dataset (language). We can observe that the NER model with endogenous augmentations achieves the better performance and holds more stable learning process under the low-resource setting. In addition, we can see that the NER model without endogenous augmentations has caused the overfitting problem due to the less training data. For example, the gap between the red solid and dashed lines increases after 5000 training iterations for English (En) dataset. The main reason is that endogenous augmentations have alleviated the overfitting issue because it searches for the diverse semantic directions and utilizes more meaningful semantic features surrounding the low-resource samples. The endogenous data operation drives the NER model to capture more generalized features within each class for effective and robust training.

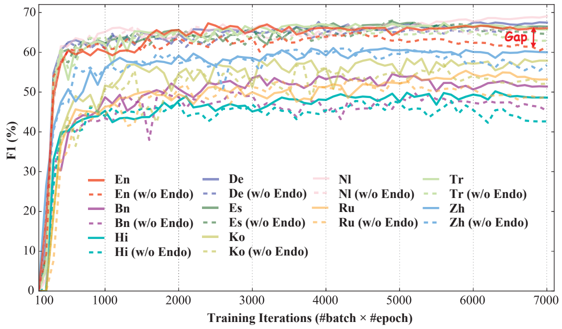


Figure 4: Performance changes as the training process goes (K=100) with/without *Endo*genous Augmentation. Total iterations equal the number of batches (#batch) × epochs (#epoch).

4.4.3 Performance Gain of Exogenous and Endogenous Augmentation. To analyze the contributions of exogenous and endogenous data augmentation to performance gains, we respectively give the test F1 scores of using only endogenous data augmentation (DA) and further introducing exogenous data augmentation (DA) under four low-resource settings (100, 200, 500, 1000) in Table 4. Compared to the previous state-of-the-art method ACLM [14], only endogenous data augmentation method has achieved optimal results on almost all datasets, respectively leading to an average increase of 4.75%, 1.98%, 1.83%, and 1.22% with 100, 200, 500, and 1000 training sentences. It is worth noting that our endogenous DA method does not require any additional explicit text data and only explores a small amount of data samples in the implicit semantic space by minimizing the upper bound of the derived loss. However, ACLM [14] retrains the language model for explicitly generating the text data and then trains the NER model based on these generated data. Overall, the endogenous augmentation process is more effective and efficient. Furthermore, we introduce the exogenous data augmentation to explicitly generate the more diverse data by querying the off-the-shelf LLM with the tailor-designed instructions. And then, the F1 score further increases by an average of 3.09%, 1.76%, 1.43%, and 1.37% under four low-resource settings. Overall, combining the exogenous and endogenous augmentations exerts a more potent effect by comprehensively considering the intrinsic semantic features and external task-related data, which significantly improves the generalization of local semantics and global text samples.

4.4.4 Diversity Analysis of the Generated Data in Exogenous Augmentation. Table 5 illustrates two examples for qualitative analysis. The first one shows that the LLM has generated a similar sentence with the original low-resource training data (Ref 1), which both hold almost identical entity context patterns, leading to small information gains from the perspective of diversity. Fortunately, our self-reflection strategy successfully detects the similarity, which

Table 4: The performance of combining exogenous and endogenous data augmentations under four low-resource settings.

#Train	Method	En	Bn	Hi	De	Es	Ko	Nl	Ru	Tr	Zh	Average
100	ACLM [14]	48.76	23.09	33.53	48.80	44.14	38.35	46.22	39.48	37.20	35.12	39.47
	Endogenous DA (Ours)	55.51	28.32	36.76	54.29	51.21	46.27	51.15	41.81	38.64	38.25	44.22
	+ Exogenous DA (Ours)	56.69	35.07	44.32	56.02	52.83	47.77	53.24	44.36	40.57	42.26	**47.31**
200	ACLM [14]	54.99	38.39	40.55	53.36	49.57	44.32	53.19	43.97	39.71	39.31	45.74
	Endogenous DA (Ours)	57.34	37.44	39.75	54.95	53.92	45.46	56.29	48.25	43.60	40.16	47.72
	+ Exogenous DA (Ours)	58.17	40.15	43.26	57.73	56.44	48.14	57.74	48.98	43.66	40.54	**49.48**
500	ACLM [14]	58.31	40.26	41.48	59.35	55.69	51.56	56.31	49.40	43.57	41.23	49.72
	Endogenous DA (Ours)	60.05	38.87	43.07	63.20	57.73	52.97	59.36	51.67	46.12	42.41	51.55
	+ Exogenous DA (Ours)	61.94	41.47	43.65	63.26	58.67	53.65	61.11	53.70	46.34	46.00	**52.98**
1000	ACLM [14]	60.14	42.42	48.20	63.80	58.33	55.55	61.22	54.31	48.23	45.19	53.74
	Endogenous DA (Ours)	61.52	42.60	48.42	66.90	59.69	55.11	61.95	55.85	46.62	50.89	54.96
	+ Exogenous DA (Ours)	62.22	45.16	50.51	66.67	60.42	56.67	64.16	56.37	48.75	52.36	**56.33**

Table 5: Examples of the generated data by LLM and the self-reflection rechecking.

Reference Sentences	**Ref 1:** he also played in [mestis]_GROUPS_ for [lempäälän kisa]_GROUPS_ , [kiekko-laser]_GROUPS_ and [jokipojat]_GROUPS_ . · · · **Ref *a*:** · · ·
Generated Data (LLM)	He also performed in [the NHL]_GROUPS_ for [the Toronto Maple Leafs]_GROUPS_, [the Montreal Canadiens]_GROUPS_, and [the New York Rangers]_GROUPS_. ×
Recheck (LLM)	**Reason:** The entities in the generated sentence are correctly labeled as [the NHL]_GROUPS_, [the Toronto Maple Leafs]_GROUPS_, [the Montreal Canadiens]_GROUPS_, and [the New York Rangers]_GROUPS_. *However, the generated sentence is similar to Reference 1, which also mentions playing in different teams.* **Output: No**
Reference Sentences	**Ref 1:** one the days passed , the child would be gifted a name and a [bulla]_PRODUCT_ during a ceremony . · · · **Ref *a*:** · · ·
Generated Data (LLM)	The [sunflower]_PRODUCT_ is a beautiful flower that blooms in the [garden]_LOCATION_ during the summer season. ✓
Recheck (LLM)	**Reason:** The entities in the generated sentence are correctly labeled. The entity [sunflower] is labeled as _PRODUCT_ and the entity [garden] is labeled as _LOCATION_. *The generated sentence is not similar to any of the reference sentences.* **Output: Yes**

Table 6: Diversity statistics on the generated data of the LLM from perspectives of the entity and context.

	200		500		1000	
	Entity	Context	Entity	Context	Entity	Context
LwTR [8]	30.72	16.46	30.07	16.22	29.85	16.55
MELM [55]	**94.85**	0.0	**94.37**	0.0	**95.13**	0.0
ACLM [14]	35.64	22.48	44.12	41.16	50.10	34.84
E^{2}DA	94.29	**31.03**	92.51	**42.98**	92.06	**37.28**

enhances the diversity and utility of the generated data in the finite number of augmentations. The second example shows a positive case which will be used for expanding the low-resource data, where the LLM generates the text with significant differences from the reference sentences. Meanwhile, we gives the quantitative experiments in Table 6. Following Ghosh et al. [14], we separately calculate the average percentage of new entities and non-entity words (context) in the generated data compared with original training data. We can see that our E^{2}DA method generates more than 90% of new entities and the highest percentage of contextual non-entity words, overall superior to MELM [55] which just replaces entities and ACLM [14] which only learns from the low-resource training corpus and then generates new sentences. This confirms the advantage of exogenous augmentation which fully explores and exploits the instruction-following and self-reflection abilities of LLMs.

5 CONCLUSION AND FUTURE WORK

We propose the exogenous and endogenous data augmentation E^{2}DA where task disentanglement based NER framework serves as the base model. The exogenous augmentation expands the original low-resource data by introducing more diverse new data, relying on the impressive instruction-following ability of the LLM. The endogenous augmentation sufficiently explores the meaningful semantic directions and exploits the infinite semantic features. Two complementary data augmentations enhance the low-resource data in terms of the local semantic and global text space, achieving notable performance. For future work, the exogenous augmentation quality can be further considered in the endogenous augmentation.

ACKNOWLEDGMENTS

This work is supported by the National Key Research and Development Program of China (grant No.2021YFB3100600), and the Youth Innovation Promotion Association of CAS (grant No.2021153).

REFERENCES

[1] Sandeep Ashwini and Jinho D Choi. 2014. Targetable named entity recognition in social media. *arXiv preprint arXiv:1408.0782* (2014).

[2] David Carmel, Ming-Wei Chang, Evgeniy Gabrilovich, Bo-June (Paul) Hsu, and Kuansan Wang. 2014. ERD'14: Entity Recognition and Disambiguation Challenge. *SIGIR Forum* (2014), 63–77.

[3] Jiaao Chen, Zichao Yang, and Diyi Yang. 2020. MixText: Linguistically-Informed Interpolation of Hidden Space for Semi-Supervised Text Classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2147–2157.

[4] Shuguang Chen, Leonardo Neves, and Thamar Solorio. 2022. Style Transfer as Data Augmentation: A Case Study on Named Entity Recognition. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 1827–1841.

[5] Xiaohua Chen, Yucan Zhou, Dayan Wu, Wanqian Zhang, Yu Zhou, Bo Li, and Weiping Wang. 2022. Imagine by reasoning: A reasoning-based implicit semantic data augmentation for long-tailed classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 356–364.

[6] Tao Cheng, Hady W. Lauw, and Stelios Paparizos. 2012. Entity Synonyms for Structured Web Search. *IEEE Transactions on Knowledge and Data Engineering* (2012), 1862–1875.

[7] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 8440–8451.

[8] Xiang Dai and Heike Adel. 2020. An Analysis of Simple Data Augmentation for Named Entity Recognition. In *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, 3861–3867.

[9] 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 *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 6045–6057.

[10] Jin Ding, Hailong Sun, Xu Wang, and Xudong Liu. 2018. Entity-Level Sentiment Analysis of Issue Comments. In *Proceedings of the 3rd International Workshop on Emotion Awareness in Software Engineering*. Association for Computing Machinery, 7–13.

[11] Christiane Fellbaum. 2010. WordNet. In *Theory and applications of ontology: computer applications*. Springer, 231–243.

[12] Besnik Fetahu, Shervin Malmasi, Anjie Fang, and Oleg Rokhlenko. 2021. Gazetteer Enhanced Named Entity Recognition for Code-Mixed WebQueries. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, 1677–1681.

[13] Sreyan Ghosh, Chandra Kiran Reddy Evuru, Sonal Kumar, S Ramaneswaran, S Sakshi, Utkarsh Tyagi, and Dinesh Manocha. 2023. DALE: Generative Data Augmentation for Low-Resource Legal NLP. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 8511–8565.

[14] Sreyan Ghosh, Utkarsh Tyagi, Manan Suri, Sonal Kumar, Ramaneswaran S, and Dinesh Manocha. 2023. ACLM: A Selective-Denoising based Generative Data Augmentation Approach for Low-Resource Complex NER. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 104–125.

[15] Jiafeng Guo, Gu Xu, Xueqi Cheng, and Hang Li. 2009. Named entity recognition in query. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '09)*. Association for Computing Machinery, 267–274.

[16] Ke Han, Shaogang Gong, Yan Huang, Liang Wang, and Tieniu Tan. 2023. Clothing-Change Feature Augmentation for Person Re-Identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 22066–22075.

[17] Zhiqi Huang, Razieh Rahimi, Puxuan Yu, Jingbo Shang, and James Allan. 2021. AutoName: A Corpus-Based Set Naming Framework. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, 2101–2105.

[18] Giulio Jacucci, Pedram Daee, Tung Vuong, Salvatore Andolina, Khalil Klouche, Mats Sjöberg, Tuukka Ruotsalo, and Samuel Kaski. 2021. Entity Recommendation for Everyday Digital Tasks. *ACM Trans. Comput.-Hum. Interact.* (2021).

[19] Ziwei Ji, Tiezheng Yu, Yan Xu, Nayeon Lee, Etsuko Ishii, and Pascale Fung. 2023. Towards Mitigating LLM Hallucination via Self Reflection. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. Association for Computational Linguistics, 1827–1843.

[20] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 7871–7880.

[21] Yinghui Li, Yangning Li, Yuxin He, Tianyu Yu, Ying Shen, and Hai-Tao Zheng. 2022. Contrastive Learning with Hard Negative Entities for Entity Set Expansion. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, 1077–1086.

[22] Jian Liu, Yufeng Chen, and Jinan Xu. 2022. Low-resource ner by data augmentation with prompting. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*. 4252–4258.

[23] Linlin Liu, Bosheng Ding, Lidong Bing, Shafiq Joty, Luo Si, and Chunyan Miao. 2021. MulDA: A Multilingual Data Augmentation Framework for Low-Resource Cross-Lingual NER. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, 5834–5846.

[24] Shervin Malmasi, Anjie Fang, Besnik Fetahu, Sudipta Kar, and Oleg Rokhlenko. 2022. MultiCoNER: A Large-scale Multilingual Dataset for Complex Named Entity Recognition. In *Proceedings of the 29th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, 3798–3809.

[25] Paul McNamee, Rion Snow, Patrick Schone, and James Mayfield. 2008. Learning Named Entity Hyponyms for Question Answering. Association for Computational Linguistics, 799–804.

[26] Shekoofeh Mokhtari, Ahmad Mahmoody, Dragomir Yankov, and Ning Xie. 2019. Tagging Address Queries in Maps Search. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 9547–9551.

[27] A. Mukherjea, M. Rao, and S. Suen. 2006. A note on moment generating functions. *Statistics and Probability Letters* (2006), 1185–1189.

[28] Frédéric Piedboeuf and Philippe Langlais. 2023. Is ChatGPT the ultimate Data Augmentation Algorithm?. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. Association for Computational Linguistics, 15606–15615.

[29] 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. (2020).

[30] Xin Rong, Zhe Chen, Qiaozhu Mei, and Eytan Adar. 2016. EgoSet: Exploiting Word Ego-networks and User-generated Ontology for Multifaceted Set Expansion. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*. Association for Computing Machinery, 645–654.

[31] Christopher Sciavolino, Zexuan Zhong, Jinhyuk Lee, and Danqi Chen. 2021. Simple Entity-Centric Questions Challenge Dense Retrievers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 6138–6148.

[32] Jiaming Shen, Zeqiu Wu, Dongming Lei, Jingbo Shang, Xiang Ren, and Jiawei Han. 2017. SetExpan: Corpus-Based Set Expansion via Context Feature Selection and Rank Ensemble. In *Machine Learning and Knowledge Discovery in Databases*. Springer International Publishing, 288–304.

[33] Noah Shinn, Beck Labash, and Ashwin Gopinath. 2023. Reflexion: an autonomous agent with dynamic memory and self-reflection. *arXiv preprint arXiv:2303.11366* (2023).

[34] Xiangyan Sun, Yanghua Xiao, Haixun Wang, and Wei Wang. 2015. On conceptual labeling of a bag of words. In *Proceedings of the 24th International Conference on Artificial Intelligence*. AAAI Press, 1326–1332.

[35] Tanmay Surana, Thi-Nga Ho, Kyaw Tun, and Eng Siong Chng. 2023. CASSI: Contextual and Semantic Structure-based Interpolation Augmentation for Low-Resource NER. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. Association for Computational Linguistics, 9729–9742.

[36] Dhaval Taunk, Lakshya Khanna, Siri Venkata Pavan Kumar Kandru, Vasudeva Varma, Charu Sharma, and Makarand Tapaswi. 2023. GrapeQA: GRaph Augmentation and Pruning to Enhance Question-Answering. Association for Computing Machinery, 1138–1144.

[37] Paul Upchurch, Jacob Gardner, Geoff Pleiss, Robert Pless, Noah Snavely, Kavita Bala, and Kilian Weinberger. 2017. Deep feature interpolation for image content changes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7064–7073.

[38] Xi Wang, Hossein Rahmani, Jiqun Liu, and Emine Yilmaz. 2023. Improving Conversational Recommendation Systems via Bias Analysis and Language-Model-Enhanced Data Augmentation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. Association for Computational Linguistics, 3609–3622.

[39] Yulin Wang, Xuran Pan, Shiji Song, Hong Zhang, Gao Huang, and Cheng Wu. 2019. Implicit Semantic Data Augmentation for Deep Networks. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc.

[40] Chenxi Whitehouse, Monojit Choudhury, and Alham Aji. 2023. LLM-powered Data Augmentation for Enhanced Cross-lingual Performance. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 671–686.

[41] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, and et al. 2020. Transformers:

State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, 38–45.

[42] Aming Wu, Rui Liu, Yahong Han, Linchao Zhu, and Yi Yang. 2021. Vector-decomposed disentanglement for domain-invariant object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9342–9351.

[43] Linzhi Wu, Pengjun Xie, Jie Zhou, Meishan Zhang, Ma Chunping, Guangwei Xu, and Min Zhang. 2022. Robust Self-Augmentation for Named Entity Recognition with Meta Reweighting. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 4049–4060.

[44] Mengzhou Xia, Xiang Kong, Antonios Anastasopoulos, and Graham Neubig. 2019. Generalized Data Augmentation for Low-Resource Translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 5786–5796.

[45] Jingyun Xu and Yi Cai. 2023. Decoupled Hyperbolic Graph Attention Network for Cross-domain Named Entity Recognition. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, 591–600.

[46] Weiwen Xu, Xin Li, Yang Deng, Wai Lam, and Lidong Bing. 2023. PeerDA: Data Augmentation via Modeling Peer Relation for Span Identification Tasks. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 8681–8699.

[47] Xiaoxin Yin and Sarthak Shah. 2010. Building Taxonomy of Web Search Intents for Name Entity Queries. In *Proceedings of the 19th International Conference on World Wide Web*. Association for Computing Machinery, 1001–1010.

[48] Puxuan Yu, Zhiqi Huang, Razieh Rahimi, and James Allan. 2019. Corpus-based Set Expansion with Lexical Features and Distributed Representations. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, 1153–1156.

[49] Puxuan Yu, Razieh Rahimi, Zhiqi Huang, and James Allan. 2020. Learning to Rank Entities for Set Expansion from Unstructured Data. In *Proceedings of the 2020 ACM SIGIR on International Conference on Theory of Information Retrieval*. Association for Computing Machinery, 21–28.

[50] Xuanyu Zhang, Qing Yang, and Dongliang Xu. 2021. Combining Explicit Entity Graph with Implicit Text Information for News Recommendation. In *Companion Proceedings of the Web Conference 2021*. Association for Computing Machinery, 412–416.

[51] Xinghua Zhang, Bowen Yu, Yubin Wang, Tingwen Liu, Taoyu Su, and Hongbo Xu. 2022. Exploring Modular Task Decomposition in Cross-domain Named Entity Recognition. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, 301–311.

[52] Yunyi Zhang, Jiaming Shen, Jingbo Shang, and Jiawei Han. 2020. Empower Entity Set Expansion via Language Model Probing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 8151–8160.

[53] Yuhao Zhang and Yongliang Wang. 2023. A Query-Parallel Machine Reading Comprehension Framework for Low-resource NER. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. Association for Computational Linguistics, 2052–2065.

[54] Deyu Zhou, Jianan Wang, Linhai Zhang, and Yulan He. 2021. Implicit Sentiment Analysis with Event-centered Text Representation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 6884–6893.

[55] 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 *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2251–2262.