Unveiling the Unknown: Open-Set Entity Typing via Two-Stage Generation

Hu Chen **Binhan Yang** **Wei Shen***
DISSec, College of Computer Science,
Nankai University, Tianjin, China
{chenhu, yangbinhan}@mail.nankai.edu.cn, shenwei@nankai.edu.cn

Abstract

Conventional fine-grained entity typing (FET) operates under the closed-set assumption, wherein all classified types are limited within a predefined type taxonomy derived from a knowledge base. As the world evolves, new entities of *unknown* types inevitably emerge in open environments, falling beyond the scope of the existing type taxonomy. To deal with this problem, in this paper, we investigate a novel and critical task: open-set entity typing (OSET), which aims to not only classify entity mentions within the known type taxonomy but also detect those outside it, termed as unknown-type instances. However, owing to the lack of exposure to unknown-type instances during training, existing FET models are susceptible to misclassify them as known types, limiting their practical effectiveness for this new OSET task. Moreover, manually collecting and annotating large-scale unknown-type instances is both time-consuming and labor-intensive in open environments. To mitigate this issue, we propose a two-stage generation model that automatically produces large-scale, high-quality and diverse pseudo unknown-type instances, beneficial for the tailor-designed unified open-set classifier to effectively distinguish between known and unknown types. Furthermore, an innovative unknown-aware hierarchical contrastive learning strategy is designed to facilitate a clear delineation between closely related known types and unknown types. Extensive experiments on two newly established benchmark datasets demonstrate that our proposed framework significantly surpasses all baselines in addressing the OSET task.

1 Introduction

Fine-grained entity typing (FET), which aims to assign fine-grained types from a given type taxonomy to entity mentions within a text, plays a fundamental role in a wide range of applications including entity linking (Onoe and Durrett, 2020; Chen et al., 2020; Shen et al., 2023), question answering (Yavuz et al., 2016; Dong et al., 2015) and relation extraction (Peng et al., 2020; Yaghoobzadeh et al., 2017). Conventionally, FET operates under the **closed-set assumption** (Chen et al., 2022; Onoe et al., 2021; Pang et al., 2022; Ding et al., 2022), i.e., all classified types belong to a predefined type taxonomy provided by a specific knowledge base (KB), implying that both training and testing entity types are derived from the same type taxonomy. Meanwhile, zero-shot FET (Yuan and Downey, 2018; Zhang et al., 2020a) aims to classify entity mentions into types that have no labeled instances during training, but still requires these type names as necessary input during inference, thus also adhering to the closed-set assumption.

However, KBs are usually curated manually and inherently incomplete (Min et al., 2013), covering a limited number of entity types. Additionally, new entities of *unknown* types frequently emerge in open environments, which do not fit into the existing type taxonomy. Therefore, practical entity typing systems should be capable of not only classifying entity mentions within the known type taxonomy but also detecting those outside it, termed as unknown-type instances. For this purpose, we propose a new task called open-set entity typing (OSET), in which testing data contain both known and unknown types while training data only have known types. Due to the lack of supervision signals for unknown types, previous closed-set FET models (Chen et al., 2022; Onoe et al., 2021; Pang et al., 2022; Ding et al., 2022) are susceptible to misclassify unknown-type instances as known types, limiting their practical feasibility. Consequently, it is vital to obtain training instances of unknown types for effectively distinguishing between known and unknown types. Unfortunately, manually collecting and annotating large-scale unknown-type instances is time-consuming and labor-intensive in open environments, creating an urgent need for automatically

*Wei Shen is the corresponding author.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 20678–20694

generating pseudo instances of unknown types.

To generate training instances automatically, various data augmentation methods (Ghosh et al., 2023; Song et al., 2024; Zhou et al., 2022) are proposed to leverage the powerful generative capabilities and extensive embodied knowledge of pre-trained language models (PLMs). Currently, a significant category of these methods is based on word-level manipulation, which produces synthetic text by leveraging the PLM's contextual knowledge to manipulate words within the original text. These methods focus on prioritizing augmentation quality but often compromising diversity and quantity. Accordingly, for the OSET task, even if this category of PLM-based methods is capable of generating high-quality and homogeneous training instances of unknown types, they fall short in producing large-scale and diverse pseudo instances (Zhang et al., 2024a). Meanwhile, large language models (LLMs) have achieved remarkable breakthroughs in producing large-scale and diverse data via in-context learning (ICL) (Chung et al., 2023; Li et al., 2023). However, recent endeavors (Lu et al., 2022; Zhao et al., 2021) have verified that ICL's performance heavily depends on the choice of demonstrations. For unknown-type instance generation, the absence of unknown-type demonstration presents a significant challenge for ICL to generalize effectively from known types to unknown types, risking unreliable and low-quality generation.

Armed with the above insights, we propose to integrate the unique strengths of PLMs and LLMs to devise a two-stage generation model, comprising a PLM-based quality-targeted generation module and an LLM-based diversity-targeted generation module. This two-stage model is expected to automatically generate large-scale, high-quality and diverse pseudo instances for unknown types, eliminating the need for human effort.

Specifically, inspired by some data augmentation methods (Zheng et al., 2020; Marek et al., 2021), high-quality pseudo unknown-type instances might exhibit surface similarities with those of known types but do not belong to any specific known type, thereby assisting in a clear delineation between known and unknown types. Based on this idea, we design a PLM-based quality-targeted generation module, which reconstructs known-type instances to derive pseudo unknown-type instances for training. Concretely, previous studies (Zhang et al., 2020b; Li et al., 2020) discover the importance of specific keywords within a textual instance in representing its semantics, and substituting these keywords can significantly change the semantics of the entire instance. Resorting to this idea, we develop a type taxonomy enhanced keyword extractor to first locate type-related keywords within known-type instances, and then substitute them with suitable candidate terms suggested by a masked language model (MLM) to yield pseudo high-quality unknown-type instances.

Building on these obtained high-quality pseudo unknown-type instances, we follow the ICL paradigm to devise an LLM-based diversity-targeted generation module, which aims to produce large-scale and diverse pseudo unknown-type instances. To select suitable demonstrations, an intuitive approach is to randomly sample in-context examples or choose the top-ranked ones. Nonetheless, these methods only treat each example independently, neglecting their inter-relationships. As a result, they may fail to achieve the desired diversity in the generated pseudo instances. To capture the interactions among examples, we exploit the determinantal point process (DPP) (Kulesza et al., 2012) to construct the most various yet informative set of demonstrations, aiding the LLM in generating diverse pseudo unknown-type instances at scale.

Finally, utilizing the pseudo unknown-type instances derived by the above two generation modules, we optimize a tailor-designed unified open-set classifier to further enhance its performance for the OSET task. Additionally, fine-grained entity types are usually linked within a type taxonomy, leading to blurred boundaries between types within the same coarse-grained type. To promote the classifier's capability of distinguishing between closely related known types and unknown types, we design an innovative unknown-aware hierarchical contrastive learning strategy by treating *known* and *unknown* as root-level types of the taxonomy and constraining the scope of attention to different hierarchical levels.

Our contributions are summarized as follows: (1) To the best of our knowledge, we are the first to investigate the task of open-set entity typing (OSET), a new and crucial problem due to its practical applications. (2) To compensate for the lack of unknown-type training instances, we develop a novel two-stage generation model, which can automatically produce large-scale, high-quality and diverse pseudo unknown-type instances. (3) To facilitate a clear delineation between closely related known types and unknown types, an innovative

unknown-aware hierarchical contrastive learning strategy is proposed to promote the distinguishing capability of the unified open-set classifier. (4) We establish two benchmark datasets for this new OSET task. Extensive experiments on these datasets demonstrate that our framework significantly outperforms state-of-the-art baselines.

2 Preliminary and Notations

The fine-grained entity typing (FET) task aims to assign each entity mention m a set of fine-grained types based on its context c given a predefined type taxonomy $\mathcal{T} = \{t_j\}_{j=1}^{|\mathcal{T}|}$. Formally, the training set $\mathcal{M} = \{(m_i, c_i)\}_{i=1}^{\mathcal{N}}$ consists of $\mathcal{N}$ entity mentions along with their corresponding contexts, where each entity mention m_i with its context c_i is labeled with a type path $\mathbf{y}_i$, which corresponds to a hierarchical path from the top level to the bottom level of the type taxonomy $\mathcal{T}$. This type path $\mathbf{y}_i$ can be represented as a set of types $\{y_{i,l}\}_{l=1}^{T} \subseteq \mathcal{T}$, where T denotes the maximum depth of the taxonomy $\mathcal{T}$ and $y_{i,l}$ signifies the type label at l-th level L_l of the type path. Given this training set $\mathcal{M}$, the closed-set FET model predicts an appropriate type path $\mathbf{y} \subseteq \mathcal{T}$ for each testing entity mention m according to its context c. While in a more realistic open-set scenario, some testing entity mentions may come from unknown types that do not belong to the predefined type taxonomy $\mathcal{T}$. Therefore, a practical model for the OSET task should not only classify entity mentions within the given type taxonomy $\mathcal{T}$, but also detect those belonging to unknown types outside of $\mathcal{T}$.

3 The Overall Framework

Figure 1 illustrates the overall architecture of our framework, which comprises three components: (1) a unified open-set classifier (§3.1); (2) a two-stage generation model (§3.2) composed of a PLM-based quality-targeted generation module (§3.2.1) and an LLM-based diversity-targeted generation module (§3.2.2); and (3) classifier optimization (§3.3).

3.1 Unified Open-Set Classifier

We design a unified open-set classifier tailored for the task of OSET, which integrates a traditional closed-set classifier $F(\cdot)$ with additional $|\mathcal{T}|$ one-vs-all (OVA) binary classifiers $\{G_{t_j}(\cdot)\}_{j=1}^{|\mathcal{T}|}$. Specifically, each OVA binary classifier $G_{t_j}(\cdot)$ is associated with a known type $t_j \in \mathcal{T}$ and produces an output $\mathbf{g}_i^{t_j} = G_{t_j}(m_i, c_i) \in \mathbb{R}^2$ for an instance (m_i, c_i), where $\mathbf{g}_i^{t_j} = \{g_i^{t_j}, \bar{g}_i^{t_j}\}$ and $g_i^{t_j} + \bar{g}_i^{t_j} = 1$. This output can be interpreted as a probability distribution, indicating the likelihood that this instance (m_i, c_i) belongs either to the known type t_j or to other types. Combining these OVA binary classifiers with the closed-set classifier, the unified open-set classifier is capable of not only classifying instances within the known type taxonomy but also detecting instances that belong to unknown types. Specifically, for an instance (m_i, c_i), the closed-set classifier $F(\cdot)$ is first utilized to obtain its known type prediction $\hat{y}_i = \text{argmax}_{t_j \in \mathcal{T}} F(m_i, c_i)$. Subsequently, the OVA binary classifier $G_{\hat{y}_i}(\cdot)$ corresponding to the predicted known type $\hat{y}_i$ is applied to output the probability $g_i^{\hat{y}_i}$, which represents the likelihood that the instance (m_i, c_i) belongs to the known type $\hat{y}_i$. If this probability $g_i^{\hat{y}_i}$ falls below 0.5, this instance is considered less likely to belong to the predicted known type $\hat{y}_i$, but regarded as unknown types; otherwise, it is classified as the predicted known type $\hat{y}_i$.

However, in the OSET task, only the known-type training set $\mathcal{M}$ are directly available for training, which inherently limits the performance of the unified open-set classifier, as demonstrated in our experiments. To overcome this limitation, we design a two-stage generation model to derive large-scale, high-quality and diverse pseudo unknown-type instances automatically. These pseudo instances are then exploited to enhance the classifier's capability of handling unknown types, thereby improving the overall performance.

3.2 Two-Stage Generation

3.2.1 PLM-Based Quality-Targeted Generation Module

To produce high-quality pseudo unknown-type instances, we propose a PLM-based quality-targeted generation module. Concretely, we first devise a type taxonomy enhanced keyword extractor to identify type-related keywords within known-type instances and then iteratively substitute these keywords with suitable candidate terms until synthesizing high-quality unknown-type instances.

Keyword Extractor. To identify the most informative keywords within a known-type instance that indicates its corresponding type path, we develop a novel type taxonomy enhanced keyword extractor, as shown in Figure 1 (a). Given an instance (m_i, c_i) from the known-type training set $\mathcal{M}$, we first concatenate it with two special tokens, `[CLS]` and `[SEP]`, to construct the input sequence $x_i = [\texttt{CLS}]m_i[\texttt{SEP}]c_i[\texttt{SEP}]$. Subsequently, a PLM

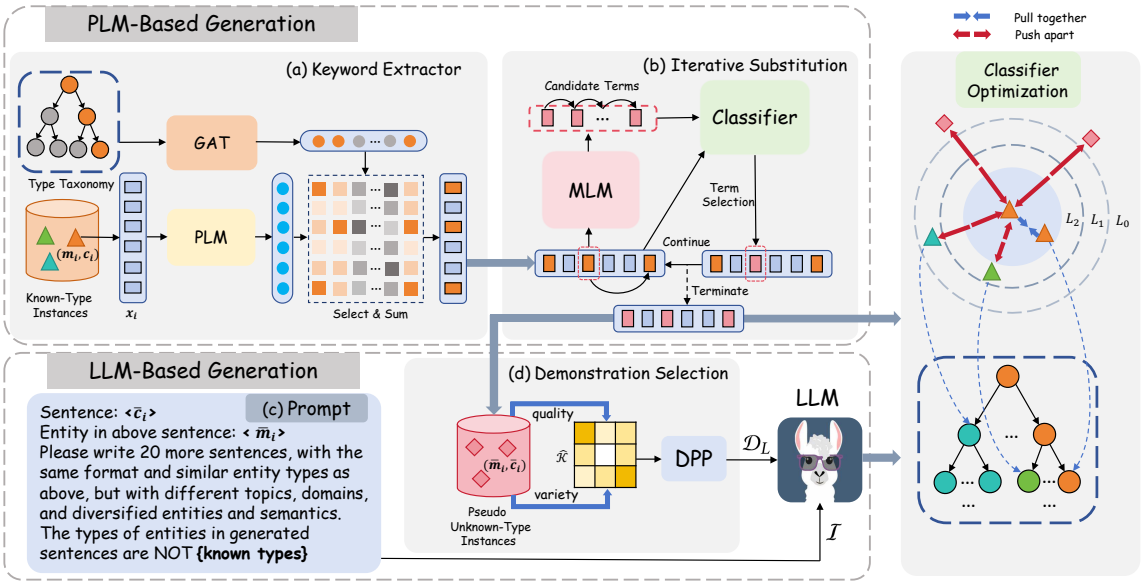


Figure 1: The overall architecture.

(e.g., BERT (Devlin et al., 2019)) is employed as the text encoder to derive the textual representation of each token in the sequence: $\mathbf{h}_{i,1}, \mathbf{h}_{i,2}, ..., \mathbf{h}_{i,n} = \text{PLM}(x_i)$. Here, $\mathbf{h}_{i,r}$ is the representation of the token $x_{i,r}$ (i.e., the r-th token of x_i). We extract the representation of the `[CLS]` token (i.e., $\mathbf{h}_{i,1}$) as the representation for the instance (m_i, c_i).

On the other hand, interpreting fine-grained type labels solely based on their type names is challenging due to the lack of sufficient semantic context. To address this, we employ the attribute value representation (AVR) module from PEARL (Liu et al., 2023), which discovers an informative seed word set for each type from the training set. With these seed word sets, AVR could generate semantics-enhanced type representations $S = \{\mathbf{s}_j\}_{j=1}^{|\mathcal{T}|}$ for entity types in $\mathcal{T}$ by computing weighted averaging representations of seed word sets. Moreover, given that the entity type set inherently forms a hierarchical structure, it is crucial to integrate both semantic and structural information into the type representation. To tackle this, we first take the obtained semantics-enhanced type representations S as initial node representations, and then leverage graph attention network (GAT) (Veličković et al., 2018) to derive the fused type representations $\text{U}=\{\mathbf{u}_j\}_{j=1}^{|\mathcal{T}|}$.

Subsequently, based on the token and fused type representations, we estimate the relatedness degree $\pi_{i,r,j} = \cos(\mathbf{h}_{i,r}, \mathbf{u}_j)$ to quantify how closely a token $x_{i,r}$ is related to a specific type t_j, where $\mathbf{h}_{i,r}$ is the representation of token $x_{i,r}$ and $\mathbf{u}_j$ is the fused type representation of type t_j. Then, the type-related score $A_{i,r}$ for token $x_{i,r}$ in the sequence x_i is calculated by summing the relatedness degrees across all types in its labeled type path $\mathbf{y}_i$: $A_{i,r} = \sum_{t_j \in \mathbf{y}_i} \text{gumbel\_softmax}(\pi_{i,r,1}, ..., \pi_{i,r,|\mathcal{T}|})_j$. According to the obtained type-related scores, we could identify type-related keywords whose scores exceed a threshold δ while masking the other tokens in the sequence x_i with `[MASK]` tokens to form a new sequence $\bar{x}_i$. Furthermore, the derived sequence $\bar{x}_i$, comprising the most informative keywords, is expected to indicate the same labeled types as the original sequence x_i. To ensure this, we train the keyword extractor using the conventional entity typing loss that enforces the two sequences x_i and $\bar{x}_i$ to have the same prediction type. After training, the final set of type-related keywords w_i are identified by the trained keyword extractor.

Iterative Substitution. High-quality pseudo unknown-type instances tend to retain surface similarities with their original known-type counterparts (Marek et al., 2021). Drawing from this observation, we propose an iterative substitution method to synthesize a high-quality pseudo unknown-type instance $(\bar{m}_i, \bar{c}_i)$ in a greedy manner, while keeping it as similar as possible to its original known-type instance (m_i, c_i), as illustrated in Figure 1 (b).

Initially, we rank all the keywords in the final type-related keyword set w_i according to their type-related scores, with the keyword having the k-th highest type-related score denoted as $w_{i,k}$. In the k-th iteration, with respect to the keyword $w_{i,k}$, candidate terms are generated based on the substituted instance from the previous $(k-1)$-th iteration. Suitable candidate terms are expected to exhibit contextual awareness in the keyword position, en-

suring that the substituted instances maintain both linguistic fluency and grammatical accuracy. Accordingly, we leverage a masked language model (MLM) to generate a set of substitution candidate terms $\mathcal{C}$ for the keyword $w_{i,k}$, thereby capitalizing on its strong context-aware prediction capabilities. The detailed substitution candidate term generation is described in Appendix A.

By replacing the keyword $w_{i,k}$ with each candidate term in $\mathcal{C}$, we can obtain $|\mathcal{C}|$ substituted instances at the k-th iteration. To identify the high-quality substituted instance within them, we introduce an unknown-type probability to evaluate the quality of the substituted instance, following the intuition that a substituted instance with a higher unknown-type probability is of higher quality. Specifically, we obtain the unknown-type probability $\bar{g}_i^{\hat{y}_i} = 1 - g_i^{\hat{y}_i}$ for a substituted instance via the unified open-set classifier (§3.1). The substituted instance with the highest unknown-type probability $\bar{g}_i^{\hat{y}_i}$ at the k-th iteration is selected. When its unknown-type probability $\bar{g}_i^{\hat{y}_i}$ exceeds 0.5, the substitution iteration will be terminated, and the corresponding substituted instance is recognized as a high-quality pseudo unknown-type instance $(\bar{m}_i, \bar{c}_i)$. Otherwise, the substitution process will continue to replace the next keyword $w_{i,k+1}$, until a high-quality pseudo unknown-type instance $(\bar{m}_i, \bar{c}_i)$ is synthesized. Ultimately, through iterative substitutions across all known-type instances in $\mathcal{M}$, we obtain a set of high-quality pseudo unknown-type instances $\mathcal{M}_P = \{(\bar{m}_i, \bar{c}_i)\}$. Note that the unified open-set classifier used in this process is trained solely with known-type instances from $\mathcal{M}$, which will be further detailed in §3.3.

3.2.2 LLM-Based Diversity-Targeted Generation Module

Building on these derived pseudo unknown-type instances $\mathcal{M}_P$, we devise an LLM-based diversity-targeted generation module that incorporates a tailor-designed prompt $\mathcal{I}$ (Figure 1 (c)) along with demonstrations $\mathcal{D}_L$ (Figure 1 (d)) selected from $\mathcal{M}_P$, to facilitate the creation of large-scale and diverse pseudo unknown-type instances.

Demonstration Selection. An intuitive strategy for selecting demonstrations is to randomly sample in-context examples or choose the top-ranked ones based on certain metrics. However, these methods tend to treat each example independently, ignoring the variety of the selected examples. Accordingly, they may inevitably limit the diversity of the generated instances, resulting in suboptimal performance. By regarding pseudo unknown-type instances $\mathcal{M}_P$ as the candidate instance pool, we aim to select e candidate instances $(\bar{m}_i, \bar{c}_i) \in \mathcal{M}_P$ to form the informative and various demonstrations $\mathcal{D}_L = \{(\bar{m}_i, \bar{c}_i)\}_{i=1}^{e}$. The selected demonstrations are desired to satisfy two key criteria: *quality* and *variety*. Firstly, based on the aforementioned insight in §3.2.1, a candidate instance with a higher unknown-type probability is generally of higher quality. Therefore, we quantify the *quality* of a candidate instance $(\bar{m}_i, \bar{c}_i)$ by computing its unknown-type probability $\bar{g}_i^{\hat{y}_i}$ via the unified open-set classifier. Secondly, *variety* requires that the selected demonstrations are mutually dissimilar, thus covering a broader range of diversified semantics within the candidate instance pool $\mathcal{M}_P$. Consequently, for two candidate instances $(\bar{m}_i, \bar{c}_i)$ and $(\bar{m}_j, \bar{c}_j)$, we calculate the cosine similarity $\phi_{i,j} = \cos(\mathbf{h}_{i,1}, \mathbf{h}_{j,1})$ to assess their semantic similarity.

To select demonstrations meeting the above two criteria, we employ the determinantal point process (DPP) (Kulesza et al., 2012), a probabilistic model renowned for its effectiveness in modeling the distribution on candidate instance sets, to derive a set of demonstrations $\mathcal{D}_L$ that is both informative and various. The primary challenge of applying DPP lies in constructing an appropriate kernel matrix. Therefore, we compute a kernel matrix $\mathbf{K} \in \mathbb{R}^{|\mathcal{M}_P| \times |\mathcal{M}_P|}$ that simultaneously accounts for *quality* and *variety*: $\mathbf{K}_{i,j} = \bar{g}_i^{\hat{y}_i} \phi_{i,j} \bar{g}_j^{\hat{y}_j}$, where $\mathbf{K}_{i,j}$ represents the possibility that candidate instances $(\bar{m}_i, \bar{c}_i)$ and $(\bar{m}_j, \bar{c}_j)$ appear in the same selected set. Intuitively, different tasks may exhibit different preferences between *quality* and *variety*. To facilitate the balance between *quality* and *variety*, we introduce a control parameter λ to construct a modified kernel matrix $\hat{\mathbf{K}}$:

$$\hat{\mathbf{K}}_{i,j} = \exp(\bar{g}_i^{\hat{y}_i}/2\lambda) \phi_{i,j} \exp(\bar{g}_j^{\hat{y}_j}/2\lambda)$$

Based on the modified kernel matrix $\hat{\mathbf{K}}$, the probability of selecting a candidate instance set $\mathcal{D}$ of size n can be defined as follows:

$$\mathcal{P}(\mathcal{D}) = \det(\hat{\mathbf{K}}_{\mathcal{D}}) / \det(\hat{\mathbf{K}} + \mathbf{I})$$

where $\hat{\mathbf{K}}_{\mathcal{D}} \in \mathbb{R}^{n \times n}$ refers to a submatrix of $\hat{\mathbf{K}}$ corresponding to the candidate instance set $\mathcal{D}$, $\mathbf{I}$ is an identity matrix and $\det(\cdot)$ denotes the determinant of a matrix. Under this distribution $\mathcal{P}$, we can search a candidate instance set $\mathcal{D}$ with the highest probability from the candidate instance pool $\mathcal{M}_P$

to obtain the demonstrations $\mathcal{D}_L$.

Despite the advantage of LLMs in generating promising pseudo instances, they might still produce plausible yet factually unreliable instances, a phenomenon known as hallucination (Dhuliawala et al., 2024). To address this, similar to §3.2.1, we adopt the unified open-set classifier to prune unreliable pseudo instances whose unknown-type probabilities $\bar{g}_i^{\hat{y}_i}$ fall below 0.5. Ultimately, we can instruct the LLM to produce a set of large-scale and diverse pseudo unknown-type instances $\mathcal{M}_L$.

3.3 Classifier Optimization

To achieve a clear delineation between known and unknown types, we utilize the set of pseudo unknown-type instances $\mathcal{M}_U = \mathcal{M}_P \cup \mathcal{M}_L$ to optimize the unified open-set classifier. Moreover, an unknown-aware hierarchical contrastive learning strategy is designed to eliminate the blurred boundaries between closely related known types and unknown types.

3.3.1 Closed-Set Classifier Optimization

Following previous FET studies (Zhang et al., 2021; Chen et al., 2019), the prediction of the type path $\mathbf{y}_i$ for an instance (m_i, c_i) can be redefined as predicting the type label $y_{i,T}$ at the bottom level L_T of the type path $\mathbf{y}_i$, capitalizing on the hierarchical dependency within $\mathcal{T}$. The closed-set classifier $F(\cdot)$ is responsible for assigning instances to predefined known types within the taxonomy. To optimize its performance, we minimize the standard cross-entropy loss over the known-type training set $\mathcal{M} = \{(m_i, c_i)\}$, formulated as follows:

$$\mathcal{L}_F = \frac{1}{|\mathcal{M}|} \sum_{(m_i, c_i) \in \mathcal{M}} \text{cross\_entropy}(y_{i,T}, \hat{y}_i)$$

where $\hat{y}_i$ is the predicted known type for the instance (m_i, c_i) obtained from $F(\cdot)$.

3.3.2 OVA Binary Classifier Optimization

Each OVA binary classifier $G_{t_j}(\cdot)$ is responsible for determining whether an instance (m_i, c_i) belongs to a specific known type t_j (with a known-type probability $g_i^{t_j}$) or other types. During its training process, for each known-type instance (m_i, c_i) with its labeled bottom-level type $y_{i,T}$ from the training set $\mathcal{M}$, the corresponding OVA binary classifier $G_{y_{i,T}}(\cdot)$ naturally treats this instance as a positive instance, thereby maximizing its known-type probability $g_i^{y_{i,T}}$. Meanwhile, any other OVA binary classifier $G_{t_j}(\cdot)$ $(t_j \in \mathcal{T} \wedge t_j \neq y_{i,T})$ is trained to minimize its corresponding known-type probability $g_i^{t_j}$ (i.e., maximizing the probability $1 - g_i^{t_j}$) by regarding this instance as a negative instance. Moreover, to endow these classifiers with the ability to distinguish between known and unknown types, each pseudo unknown-type instance $(\bar{m}_i, \bar{c}_i)$ from the set of pseudo instances $\mathcal{M}_U$ is also regarded as a negative instance for every OVA binary classifier $G_{t_j}(\cdot)$ to minimize its known-type probability $g_i^{t_j}$. Armed with this insight, the optimization of these OVA binary classifiers could be achieved by the following formula:

$$\mathcal{L}_G = \frac{1}{|\mathcal{M}|} \sum_{(m_i, c_i) \in \mathcal{M}} (-\log(g_i^{y_{i,T}}) - \min_{t_j \in \mathcal{T} \wedge t_j \neq y_{i,T}} \log(1 - g_i^{t_j})) + \frac{1}{|\mathcal{M}_U|} \sum_{(\bar{m}_i, \bar{c}_i) \in \mathcal{M}_U} - \min_{t_j \in \mathcal{T}} \log(1 - g_i^{t_j})$$

3.3.3 Unknown-Aware Hierarchical Contrastive Learning

Fine-grained known types within a type taxonomy are typically interconnected, making it challenging to differentiate them, especially those within the same coarse-grained type. Inspired by previous contrastive learning methods (He et al., 2020; Wang et al., 2022), we design an unknown-aware hierarchical contrastive learning strategy that eliminates confusion among intertwined known types while enhancing the capability of the unified open-set classifier to distinguish known and unknown types. To be specific, for each known-type instance (m_i, c_i) with its labeled type path $\mathbf{y}_i$ from the training set $\mathcal{M}$, we sample positive instances that share the same labeled type path $\mathbf{y}_i$, thereafter enforcing them to be close together in the latent representation space. What's more, to facilitate the classifier to receive sufficient supervision signals from all hierarchical levels of the labeled type path $\mathbf{y}_i$, we sample negative instances from the training set $\mathcal{M}$ at each level L_l $(1 \leq l \leq T)$. Specifically, each negative instance at level L_l is expected to have a different type label $y_{i,l}$ but share the same higher-level type labels $\{y_{i,z}\}_{z=1}^{l-1}$. Additionally, to further strengthen the classifier's capability of distinguishing between known and unknown types, we treat *known* and *unknown* as types at the root level (defined as L_0) of the type taxonomy $\mathcal{T}$, which enables the sampling of negative instances at the level L_0 from the set of pseudo unknown-type instances $\mathcal{M}_U$. During the sampling process, for each instance (m_i, c_i), we select Q least similar positive instances to form its positive set $P(i)$ and Q most similar negative instances to form its negative set

Methods	BBN				Few-NERD			
	Acc-all	F1-all	F1-known	F1-binary	Acc-all	F1-all	F1-known	F1-binary
Fine-Grained Entity Typing								
UFET	0.384	0.518	0.390	0.717	0.267	0.460	0.375	0.728
Box4Types	0.598	0.642	0.771	0.730	0.475	0.564	0.670	0.712
UniST	0.279	0.414	0.481	0.672	0.170	0.388	0.448	0.676
CASENT	0.295	0.444	0.703	0.666	0.070	0.416	0.613	0.669
ChatGPT	0.489	0.652	0.786	0.724	0.453	0.486	0.465	0.657
Llama3	0.298	0.390	0.537	0.569	0.387	0.520	0.528	0.672
DeepSeek-V3	0.622	0.688	0.716	0.668	0.418	0.360	0.365	0.690
Open-Set Text Classification								
MSP	0.556	0.573	0.720	0.707	0.583	0.640	0.661	0.741
SEG	0.455	0.446	0.563	0.662	0.519	0.575	0.557	0.750
SCL	0.727	0.758	0.786	0.774	0.413	0.476	0.569	0.688
APRL	0.616	0.640	0.731	0.734	0.567	0.625	0.654	0.726
SELFSUP	0.675	0.700	0.753	0.774	0.525	0.591	0.641	0.734
Ours	**0.763**	**0.811**	**0.851**	**0.823**	**0.621**	**0.675**	**0.687**	**0.770**

Table 1: Results on BBN and Few-NERD. The best result in each metric is in **bold** and the second-best is underlined.

$N_l(i)$ at each level L_l $(0 \leq l \leq T)$, ensuring the focus on difficult positive and negative instances. Therefore, the unknown-aware hierarchical contrastive loss $\mathcal{L}_H$ is defined as follows:

$$\mathcal{L}_H = \frac{1}{|\mathcal{M}|} \sum_{(m_i, c_i) \in \mathcal{M}} \frac{1}{|P(i)|} \sum_{(m_{i^+}, c_{i^+}) \in P(i)} \sum_{l=0}^{T} -\log \frac{\exp\left(\frac{\mathbf{h}_{i,1} \cdot \mathbf{h}_{i^+,1}}{\tau}\right)}{\exp\left(\frac{\mathbf{h}_{i,1} \cdot \mathbf{h}_{i^+,1}}{\tau}\right) + \sum_{(m_{i^-}, c_{i^-}) \in N_l(i)} \exp\left(\frac{\mathbf{h}_{i,1} \cdot \mathbf{h}_{i^-,1}}{\tau}\right)}$$

Here, τ is a temperature parameter that controls the strength of contrastive learning.

By incorporating the above three losses, the overall objective function can be formulated as follows:

$$\mathcal{L} = \mathcal{L}_F + \mathcal{L}_G + \alpha \mathcal{L}_H$$

where α controls the strength of $\mathcal{L}_H$.

4 Experiments

4.1 Datasets and Experiment Setting

4.1.1 Datasets and Metrics

To the best of our knowledge, there is no publicly available benchmark dataset for the OSET task. Therefore, we create two benchmarks by adapting existing fine-grained entity typing datasets: BBN (Weischedel and Brunstein, 2005) and Few-NERD (Ding et al., 2021). The construction process and setting details are elaborated in Appendices B and C, respectively. Following prior open-set text classification studies (Zhan et al., 2021; Zeng et al., 2021), we regard all unknown types as a single type and adopt four widely used metrics for evaluation: (1) **Acc-all**, the accuracy over all types; (2) **F1-all**, the weighted-F1 over all types; (3) **F1-known**, the weighted-F1 score for known types; and (4) **F1-binary**, a binary classification metric that assesses the ability to detect unknown-type instances. The datasets and source code are available for future research [1].

Variants	BBN			
	Acc-all	F1-all	F1-known	F1-binary
FULL	**0.763**	**0.811**	**0.851**	**0.823**
PGM	$0.683_{\downarrow 8.0\%}$	$0.739_{\downarrow 7.2\%}$	$0.817_{\downarrow 3.4\%}$	$0.768_{\downarrow 5.5\%}$
LGM	$0.574_{\downarrow 18.9\%}$	$0.620_{\downarrow 19.1\%}$	$0.772_{\downarrow 7.9\%}$	$0.717_{\downarrow 10.6\%}$
BASE	$0.562_{\downarrow 20.1\%}$	$0.612_{\downarrow 19.9\%}$	$0.786_{\downarrow 6.5\%}$	$0.708_{\downarrow 11.5\%}$

Variants	Few-NERD			
	Acc-all	F1-all	F1-known	F1-binary
FULL	**0.621**	**0.675**	**0.687**	**0.770**
PGM	$0.573_{\downarrow 4.8\%}$	$0.636_{\downarrow 3.9\%}$	$0.667_{\downarrow 2.0\%}$	$0.745_{\downarrow 2.5\%}$
LGM	$0.507_{\downarrow 11.4\%}$	$0.576_{\downarrow 9.9\%}$	$0.659_{\downarrow 2.8\%}$	$0.724_{\downarrow 4.6\%}$
BASE	$0.488_{\downarrow 13.3\%}$	$0.555_{\downarrow 12.0\%}$	$0.650_{\downarrow 3.7\%}$	$0.714_{\downarrow 5.6\%}$

Table 2: Results of different variants of our framework.

4.1.2 Baselines

To our best knowledge, no prior work has been proposed for solving the OSET task. For comprehensive comparison, we establish twelve baselines based on fine-grained entity typing methods (i.e., UFET (Choi et al., 2018), Box4Types (Onoe et al., 2021), UniST (Huang et al., 2022), CASENT (Feng et al., 2023), ChatGPT (Ouyang et al., 2022), Llama3 (AI@Meta, 2024) and DeepSeek-V3 (Liu et al., 2024)) and open-set text classification methods (i.e., MSP (Hendrycks and Gimpel, 2017), SEG (Yan et al., 2020), SCL (Zeng et al., 2021), APRL (Chen et al., 2021) and SELFSUP (Zhan et al., 2021)), which are introduced in Appendix D.

4.2 Performance Comparison

The experimental results of all methods over BBN and Few-NERD datasets are shown in Table 1. From the results, it can be seen that our framework achieves significantly superior performance compared with these fine-grained entity typing (FET) and open-set text classification (OSTC) baselines in terms of all metrics on both datasets. To be specific, there is a significant performance gap between our framework and FET baselines, demon-

[1] https://github.com/CodingPerson/OSET

Method	BBN				Few-NERD			
	Acc-all	F1-all	F1-known	F1-binary	Acc-all	F1-all	F1-known	F1-binary
Ours	**0.763**	**0.811**	**0.851**	**0.823**	**0.621**	**0.675**	**0.687**	**0.770**
(1) w/o KE	$0.680_{\downarrow 8.3\%}$	$0.731_{\downarrow 8.0\%}$	$0.801_{\downarrow 5.0\%}$	$0.777_{\downarrow 4.6\%}$	$0.609_{\downarrow 1.2\%}$	$0.665_{\downarrow 1.0\%}$	$0.680_{\downarrow 0.7\%}$	$0.760_{\downarrow 1.0\%}$
(2) w/o DS	$0.720_{\downarrow 4.3\%}$	$0.783_{\downarrow 2.8\%}$	$0.850_{\downarrow 0.1\%}$	$0.791_{\downarrow 3.2\%}$	$0.535_{\downarrow 8.6\%}$	$0.627_{\downarrow 4.8\%}$	$0.669_{\downarrow 1.8\%}$	$0.737_{\downarrow 3.3\%}$
(3) w/o UHCL	$0.703_{\downarrow 6.0\%}$	$0.740_{\downarrow 7.1\%}$	$0.794_{\downarrow 5.7\%}$	$0.787_{\downarrow 3.6\%}$	$0.601_{\downarrow 2.0\%}$	$0.665_{\downarrow 1.0\%}$	$0.686_{\downarrow 0.1\%}$	$0.763_{\downarrow 0.7\%}$

Table 3: Experimental results of ablation study.

strating the effectiveness of our framework generating effective pseudo unknown-type instances to enhance the detection of unknown types. While LLMs (i.e., ChatGPT, Llama3, and DeepSeek-V3) have shown remarkable capabilities across various downstream tasks, they still face challenges in tackling the OSET task, which may be attributed to the lack of sufficient prior knowledge for this new task. In comparison to FET baselines, the OSTC baselines (i.e., MSP, SEG, SCL, APRL, and SELFSUP) could learn discriminative semantic representations using known-type instances to detect unknown-type instances, with SCL and SELFSUP further enhancing their performance by synthesizing pseudo instances during training. However, the performance of these OSTC baselines is still unsatisfactory as they inherently neglect the hierarchical structure of the type taxonomy. In contrast, our framework is capable of producing desirable pseudo instances while also capturing the hierarchical structure of the type taxonomy, thereby providing a promising way to address the OSET task.

4.3 Effectiveness Study of Two-Stage Generation

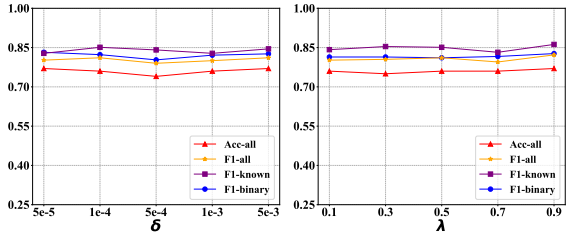
To validate the effectiveness of our proposed two-stage generation model, we conduct an extensive evaluation by considering different variants of our framework: (1) BASE, which utilizes only the labeled known-type instances for training; (2) PGM, where only the PLM-based generation module is employed to produce pseudo instances for training besides the input known-type instances; (3) LGM, where only the LLM-based generation module is leveraged to generate pseudo instances for training alongside the input known-type instances; (4) FULL (i.e., our full framework).

Firstly, it can be seen from Table 2 that BASE shows the largest performance degradation across all metrics on both datasets compared to FULL, validating the effectiveness of the pseudo unknown-type instances yielded by our two-stage generation model. Comparing PGM with FULL, we observe a noticeable performance decline for PGM. This result supports our motivation that the LLM-based generation module can facilitate the creation of diverse and varied pseudo unknown-type instances across different topics and domains, thus enhancing the classifier's performance by capturing more generalized and complementary knowledge. Additionally, LGM's performance gap w.r.t. FULL is much more significant than PGM, which implies that the high-quality pseudo unknown-type instances are more effective in addressing the OSET task than the diverse pseudo instances. Moreover, the performance of BASE is comparable to that of LGM, further confirming that our framework shows limited efficacy when relying solely on the LLM-based generation module.

Overall, combining the high-quality and diverse pseudo unknown-type instances derived from our two-stage generation model can exert a more potent effect by comprehensively integrating the unique strengths of PLMs and LLMs. Furthermore, a time cost analysis and a case study for two-stage generation are provided in Appendices E and F, respectively.

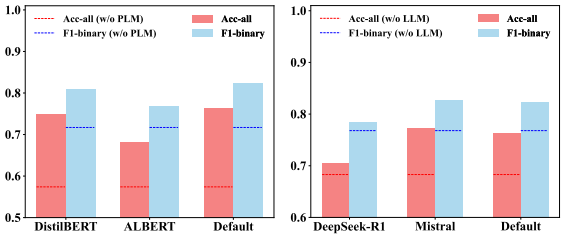
4.4 Ablation Study

To verify the superiority of key components in our framework, we conduct an extensive ablation study in Table 3. From the experimental results, we can see that: (1) By replacing the keyword extractor with random selection (w/o KE), we observe a significant performance decline compared to the whole framework. This result indicates the efficacy of our keyword extractor to precisely identify type-related keywords for substitution, thus guaranteeing the quality of generated pseudo unknown-type instances. (2) Additionally, w.r.t. the whole framework, significant performance decreases are observed when demonstrations are selected randomly (w/o DS), showcasing that the proposed demonstration selection process could indeed select satisfactory demonstrations to guide the LLM in generating valuable and diverse pseudo instances, thereby enhancing the overall performance. (3) Moreover, the removal of the unknown-aware hierarchical contrastive learning strategy (w/o UHCL) leads to performance declines over the two datasets, which confirms that the proposed strategy effectively eliminates the blurred boundaries between closely related known types and unknown types.



(a) threshold δ (b) control parameter λ

Figure 2: Hyperparameter sensitivity analysis.



(a) variants of PLMs (b) variants of LLMs

Figure 3: Performance of different variants using different PLMs and LLMs.

4.5 Hyperparameter Sensitivity Analysis

To evaluate the robustness of our framework, we conduct a thorough sensitivity analysis on BBN for two pivotal hyperparameters: the type-related keyword identification threshold δ (§3.2.1) and the control parameter λ that balances *quality* and *variety* (§3.2.2). As depicted in Figure 2 (a), our framework maintains stable performance in terms of all metrics across a broad range of δ (5e-5 to 5e-3), exhibiting the framework's robustness and insensitivity to the threshold δ. Likewise, from Figure 2 (b), we observe consistent performance across all metrics over the range of λ from 0.1 to 0.9, further confirming the framework's stability with respect to the parameter λ.

4.6 Effect Analysis of Different PLMs and LLMs

To validate the universality of our proposed two-stage generation model across diverse pretrained language models (PLMs) and large language models (LLMs), we conduct comprehensive evaluations on BBN, as shown in Figure 3.

For the PLM-based quality-targeted generation module, we instantiate three variants by incorporating the default LLM (Llama3) with three distinct PLMs: DistilBERT (Sanh et al., 2019), ALBERT (Lan et al., 2020), and BERT (default PLM). We provide their performance in terms of Acc-all and F1-binary in Figure 3 (a). Additionally, the dotted line in the figure represents the performance of the baseline without using any PLM, thus highlighting the performance gain facilitated by PLMs. From Figure 3 (a), we can draw two critical observations: (1) variants with different PLMs (DistilBERT, ALBERT, and BERT) significantly outperform the baseline, which demonstrates the effectiveness and robustness of our proposed PLM-based generation module over the different choices of PLMs. (2) The variant with BERT surpasses DistilBERT and ALBERT, likely attributed to their inherent limitations of lightweight architectures. Specifically, ALBERT and DistilBERT employ parameter reduction techniques that constrain their knowledge retention and text generation capabilities, thereby inevitably producing low-quality pseudo unknown-type instances for training.

To further examine the universality of the LLM-based diversity-targeted generation module, we integrate the default PLM (BERT) with three different LLMs: DeepSeek-R1 (Guo et al., 2025), Mistral (Jiang et al., 2023), and Llama3 (default LLM). We present the performance of these variants and a baseline without utilizing any LLM in Figure 3 (b), where the dotted line denotes the performance of the baseline. From this figure, it can be seen that (1) variants with different LLMs (DeepSeek-R1, Mistral, and Llama3) all substantially exceed the baseline in terms of Acc-all and F1-binary, which verifies the universality of our proposed LLM-based generation module over different LLMs. (2) The variant with DeepSeek-R1 yields worse performance compared to the other two variants. This can be explained by the fact that the reinforcement learning-based architecture of DeepSeek-R1 may inject extraneous reasoning steps during unknown-type instance generation, thereby inducing error propagation that ultimately causes suboptimal performance.

In summary, the superior performance of these variants validates the effectiveness and universality of our two-stage generation model across different PLMs and LLMs.

5 Conclusion

In this paper, we investigate a novel open-set entity typing task, which aims to not only classify entity mentions within the known type taxonomy but also detect those outside it, i.e., unknown-type instances. To tackle this task, we propose a two-stage generation model to produce pseudo unknown-type instances that enhance our designed unified open-set classifier. Moreover, an unknown-aware hierarchical contrastive learning strategy is introduced to clearly differentiate between closely related known types and unknown types. Experiments over two created benchmark datasets demonstrate that our framework consistently outperforms all baselines.

Limitations

Firstly, our created two benchmarks BBN and Few-NERD (by adapting existing mainstream fine-grained entity typing datasets) exhibit a long-tailed distribution, which inevitably leads to performance disparities between high-frequency head types and low-frequency tail types in our OSET task. Addressing this imbalance problem remains an important research direction for future work. Secondly, our proposed framework is designed not only to classify entity mentions within the known type taxonomy but also to detect those out of it, termed as unknown-type instances. Based on these detected unknown-type instances, it is feasible to further discover new entity types and accurately integrate them into the existing taxonomy, thereby enabling dynamic taxonomy updates. This will be a pivotal direction for advancing the practical deployment of entity typing systems in real-world scenarios.

Ethical Statement

Our proposed two-stage generation model integrates the unique strengths of pretrained language models (PLMs) and large language models (LLMs) to produce pseudo unknown-type instances, facilitating the training of a unified open-set classifier for the novel OSET task. However, it is important to acknowledge that the generated pseudo instances may unintentionally reflect societal bias embedded in the underlying knowledge of PLMs and LLMs. To mitigate potential ethical concerns, we recommend incorporating human oversight to review the generated pseudo instances, thereby ensuring the ethical and fair application of our framework in real-world scenarios.

References

AI@Meta. 2024. Llama 3 model card.

Bo Chen, Xiaotao Gu, Yufeng Hu, Siliang Tang, Guoping Hu, Yueting Zhuang, and Xiang Ren. 2019. Improving distantly-supervised entity typing with compact latent space clustering. In *NAACL-HLT*, pages 2862–2872.

Guangyao Chen, Peixi Peng, Xiangqian Wang, and Yonghong Tian. 2021. Adversarial reciprocal points learning for open set recognition. *IEEE TPAMI*, 44(11):8065–8081.

Shuang Chen, Jinpeng Wang, Feng Jiang, and Chin-Yew Lin. 2020. Improving entity linking by modeling latent entity type information. In *AAAI*, pages 7529–7537.

Yi Chen, Jiayang Cheng, Haiyun Jiang, Lemao Liu, Haisong Zhang, Shuming Shi, and Ruifeng Xu. 2022. Learning from sibling mentions with scalable graph inference in fine-grained entity typing. In *ACL*, pages 2076–2087.

Eunsol Choi, Omer Levy, Yejin Choi, and Luke Zettlemoyer. 2018. Ultra-fine entity typing. In *ACL*, pages 87–96.

John Chung, Ece Kamar, and Saleema Amershi. 2023. Increasing diversity while maintaining accuracy: Text data generation with large language models and human interventions. In *ACL*, pages 575–593.

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

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason E Weston. 2024. Chain-of-verification reduces hallucination in large language models. In *ICLR-RRFM*.

Ning Ding, Yulin Chen, Xu Han, Guangwei Xu, Xiaobin Wang, Pengjun Xie, Haitao Zheng, Zhiyuan Liu, Juanzi Li, and Hong-Gee Kim. 2022. Prompt-learning for fine-grained entity typing. In *EMNLP-Findings*, pages 6888–6901.

Ning Ding, Guangwei Xu, Yulin Chen, Xiaobin Wang, Xu Han, Pengjun Xie, Haitao Zheng, and Zhiyuan Liu. 2021. Few-nerd: A few-shot named entity recognition dataset. In *ACL-IJCNLP*, pages 3198–3213.

Li Dong, Furu Wei, Hong Sun, Ming Zhou, and Ke Xu. 2015. A hybrid neural model for type classification of entity mentions. In *IJCAI*, pages 1243–1249.

Yanlin Feng, Adithya Pratapa, and David R Mortensen. 2023. Calibrated seq2seq models for efficient and generalizable ultra-fine entity typing. In *EMNLP-Findings*, pages 15550–15560.

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 *ACL*, pages 104–125.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv*.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9729–9738.

Dan Hendrycks and Kevin Gimpel. 2017. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *ICLR*.

James Y Huang, Bangzheng Li, Jiashu Xu, and Muhao Chen. 2022. Unified semantic typing with meaningful label inference. In *NAACL-HLT*, pages 2642–2654.

Lalit P Jain, Walter J Scheirer, and Terrance E Boult. 2014. Multi-class open set recognition using probability of inclusion. In *ECCV*, pages 393–409.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, et al. 2023. Mistral 7b. *arXiv*.

Hailong Jin, Lei Hou, Juanzi Li, and Tiansi Dong. 2019. Fine-grained entity typing via hierarchical multi graph convolutional networks. In *EMNLP-IJCNLP*, pages 4969–4978.

Alex Kulesza, Ben Taskar, et al. 2012. Determinantal point processes for machine learning. *FTML*, 5(2–3):123–286.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *ICLR*.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. Bert-attack: Adversarial attack against bert using bert. In *EMNLP*, pages 6193–6202.

Zhuoyan Li, Hangxiao Zhu, Zhuoran Lu, and Ming Yin. 2023. Synthetic data generation with large language models for text classification: Potential and limitations. In *EMNLP*, pages 10443–10461.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024. Deepseek-v3 technical report. *arXiv*.

Yinan Liu, Hu Chen, Wei Shen, and Jiaoyan Chen. 2023. Low-resource personal attribute prediction from conversations. In *AAAI*, pages 4507–4515.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *ACL*, pages 8086–8098.

Petr Marek, Vishal Ishwar Naik, Anuj Goyal, and Vincent Auvray. 2021. Oodgan: Generative adversarial network for out-of-domain data generation. In *NAACL-HLT*, pages 238–245.

Bonan Min, Ralph Grishman, Li Wan, Chang Wang, and David Gondek. 2013. Distant supervision for relation extraction with an incomplete knowledge base. In *NAACL-HLT*, pages 777–782.

Yasumasa Onoe, Michael Boratko, Andrew Mccallum, and Greg Durrett. 2021. Modeling fine-grained entity types with box embeddings. In *ACL-IJCNLP*, pages 2051–2064.

Yasumasa Onoe and Greg Durrett. 2020. Fine-grained entity typing for domain independent entity linking. In *AAAI*, pages 8576–8583.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. In *NIPS*, pages 27730–27744.

Kunyuan Pang, Haoyu Zhang, Jie Zhou, and Ting Wang. 2022. Divide and denoise: Learning from noisy labels in fine-grained entity typing with cluster-wise loss correction. In *ACL*, pages 1997–2006.

Hao Peng, Tianyu Gao, Xu Han, Yankai Lin, Peng Li, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 2020. Learning from context or names? an empirical study on neural relation extraction. In *ENNLP*, pages 3661–3672.

Ryan Rifkin and Aldebaro Klautau. 2004. In defense of one-vs-all classification. *JMLR*, 5:101–141.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv*.

Wei Shen, Yuhan Li, Yinan Liu, Jiawei Han, Jianyong Wang, and Xiaojie Yuan. 2023. Entity linking meets deep learning: Techniques and solutions. *IEEE TKDE*, 35(3):2556–2578.

Lei Shu, Hu Xu, and Bing Liu. 2017. Doc: Deep open classification of text documents. In *EMNLP*, pages 2911–2916.

Sihan Song, Furao Shen, and Jian Zhao. 2024. Ropda: Robust prompt-based data augmentation for low-resource named entity recognition. In *AAAI*, pages 19017–19025.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *ICLR*.

Zihan Wang, Peiyi Wang, Lianzhe Huang, Xin Sun, and Houfeng Wang. 2022. Incorporating hierarchy into text encoder: a contrastive learning approach for hierarchical text classification. In *ACL*, pages 7109–7119.

Ralph Weischedel and Ada Brunstein. 2005. Bbn pronoun coreference and entity type corpus. *Linguistic Data Consortium, Philadelphia*.

Yadollah Yaghoobzadeh, Heike Adel, and Hinrich Schütze. 2017. Noise mitigation for neural entity typing and relation extraction. In *EACL*, pages 1183–1194.

Guangfeng Yan, Lu Fan, Qimai Li, Han Liu, Xiaotong Zhang, Xiao-Ming Wu, and Albert YS Lam. 2020. Unknown intent detection using gaussian mixture model with an application to zero-shot intent classification. In *ACL*, pages 1050–1060.

Semih Yavuz, Izzeddin Gür, Yu Su, Mudhakar Srivatsa, and Xifeng Yan. 2016. Improving semantic parsing via answer type inference. In *EMNLP*, pages 149–159.

Zheng Yuan and Doug Downey. 2018. Otyper: A neural architecture for open named entity typing. In *AAAI*, pages 6037–6044.

Zhiyuan Zeng, Keqing He, Yuanmeng Yan, Zijun Liu, Yanan Wu, Hong Xu, Huixing Jiang, and Weiran Xu. 2021. Modeling discriminative representations for out-of-domain detection with supervised contrastive learning. In *ACL-IJCNLP*, pages 870–878.

Li-Ming Zhan, Haowen Liang, Bo Liu, Lu Fan, Xiao-Ming Wu, and Albert YS Lam. 2021. Out-of-scope intent detection with self-supervision and discriminative training. In *ACL-IJCNLP*, pages 3521–3532.

Haoyu Zhang, Dingkun Long, Guangwei Xu, Muhua Zhu, Pengjun Xie, Fei Huang, and Ji Wang. 2021. Learning with noise: improving distantly-supervised fine-grained entity typing via automatic relabeling. In *IJCAI*, pages 3808–3815.

Tao Zhang, Congying Xia, Chun-Ta Lu, and S Yu Philip. 2020a. Mzet: Memory augmented zero-shot fine-grained named entity typing. In *COLING*, pages 77–87.

Wei Emma Zhang, Quan Z. Sheng, Ahoud Alhazmi, and Chenliang Li. 2020b. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM TIST*, 11(3):1–41.

Xinghua Zhang, Gaode Chen, Shiyao Cui, Jiawei Sheng, Tingwen Liu, and Hongbo Xu. 2024a. Exogenous and endogenous data augmentation for low-resource complex named entity recognition. In *SIGIR*, pages 630–640.

Yu Zhang, Yunyi Zhang, Yanzhen Shen, Yu Deng, Lucian Popa, Larisa Shwartz, ChengXiang Zhai, and Jiawei Han. 2024b. Seed-guided fine-grained entity typing in science and engineering domains. In *AAAI*, pages 19606–19614.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *ICML*, pages 12697–12706.

Yinhe Zheng, Guanyi Chen, and Minlie Huang. 2020. Out-of-domain detection for natural language understanding in dialog systems. *IEEE TASLP*, 28:1198–1209.

Jing Zhou, Yanan Zheng, Jie Tang, Li Jian, and Zhilin Yang. 2022. Flipda: Effective and robust data augmentation for few-shot learning. In *ACL*, pages 8646–8665.

Xinyu Zuo, Haijin Liang, Ning Jing, Shuang Zeng, Zhou Fang, and Yu Luo. 2022. Type-enriched hierarchical contrastive strategy for fine-grained entity typing. In *COLING*, pages 2405–2417.

A Substitution Candidate Term Generation

To replace identified keywords with appropriate candidate terms, we adopt a masked language model (MLM) that leverages its intrinsic contextual prediction capability to generate a set of substitution candidate terms.

Specifically, a keyword $w_{i,k}$ is first masked with a `[MASK]` token, and then fed into the MLM along with its surrounding context to predict possible candidate terms. The MLM yields the `[MASK]` token representation $\mathbf{h}_{\texttt{[MASK]}}$ and outputs a probability distribution over the entire vocabulary V, which indicates the likelihood of each candidate term v appearing in the masked position:

$$p_V(v|\mathbf{h}_{\texttt{[MASK]}}) = \text{softmax}(W_2\sigma(W_1\mathbf{h}_{\texttt{[MASK]}} + b))$$

Here, $\sigma(\cdot)$ is an activation function; $W_1 \in \mathbb{R}^{d\times d}$, $W_2 \in \mathbb{R}^{|V|\times d}$, and $b \in \mathbb{R}^d$ are pre-trained parameters of the MLM. Since the keyword substitution process aims to alter the type of the instance, it is crucial to avoid substituting a keyword with another known-type related term, which could inadvertently preserve the original type. To mitigate this risk, we exclude the top 10% of terms with the highest type-related scores for each known type from the vocabulary V, ensuring that highly known-type related terms are not utilized as substitution candidate terms. After this exclusion, top-200 suggested terms are selected as the substitution candidate terms $\mathcal{C}$ w.r.t. the keyword $w_{i,k}$, according to the probability distribution $p_V(v|\mathbf{h}_{\texttt{[MASK]}})$.

B Dataset Construction Details

We construct two benchmark datasets of the OSET task based on existing FET datasets (i.e., BBN (Weischedel and Brunstein, 2005) and Few-NERD (Ding et al., 2021)), where the type labels are redefined and split into known and unknown types, and the data instances are reorganized to meet the specific requirements of the OSET task, described in details as follows.

- **BBN** (Weischedel and Brunstein, 2005) is extracted from Wall Street Journal articles and annotated with 2 hierarchical levels of 47 types in Freebase, among which we designate 27 types as known types and regard the remaining 20 types as unknown. Note that the selected known types should establish a complete hierarchical structure, and the specific type allocation result is

shown in Table 5. In the light of the OSET task definition, the testing set should contain both known-type and unknown-type instances, while the training and validation sets only have known-type instances. Accordingly, for the testing set, we sample a balanced distribution of 50% known-type and 50% unknown-type instances, and the remaining known-type instances are allocated to form the training and validation sets.

- **Few-NERD** (Ding et al., 2021) is a large-scale human-annotated dataset with 2 hierarchical levels of 74 types. Similar to BBN, we divide the types into 37 known types and 37 unknown types (the specific type allocation result is also shown in Table 5), and adopt the same manner to construct its training, validation and testing set.

The statistics of these two constructed datasets are shown in Table 4.

Dataset	BBN	Few-NERD
# Known types	27	37
# Unknown types	20	37
# Training instances	7996	35193
# Validation instances	2431	31729
# Known-type testing instances	961	16176
# Unknown-type testing instances	961	16176

Table 4: Dataset statistics.

C Setting Details

To instantiate our framework, we select BERT (Devlin et al., 2019) as the default PLM and Llama3-8B (AI@Meta, 2024) as the default LLM in our experiments. For the LLM-based generation module, we select sets of demonstrations from the candidate instance pool $\mathcal{M}_P$ without replacement to perform multiple rounds of the generation process. Despite that the LLM-based generation module is capable of producing large-scale pseudo unknown-type instances, we just reserve $\mathcal{N}$ pseudo instances (the same size with the original training set $\mathcal{M}$) to form the set of pseudo unknown-type instances $\mathcal{M}_L$, thus balancing the performance of PLM-based and LLM-based generation modules without the influence of the generated instance size. The threshold δ (Section 3.2.1) is set to 1e-4. The control parameters λ (Section 3.2.2) and α (Section 3.3.3) are set to 0.5 and 0.1, respectively. The number of demonstrations e is set to 5. For the unknown-aware hierarchical contrastive learning, we set Q to 25, and the temperature parameter τ is set to 0.07. We use Adam as the optimizer with a learning rate of 3e-5. All experiments are implemented using PyTorch with one NVIDIA RTX A6000 GPU.

D Descriptions of Baseline Methods

(i) *Fine-grained entity typing.* For fine-grained entity typing methods, we incorporate a threshold mechanism that enables their identification of unknown-type instances.

- UFET (Choi et al., 2018) utilizes BiLSTM and GloVe for context encoding, while employing GloVe and CharCNN for entity mention encoding. During inference, it learns a type label matrix to estimate each type's probability for multi-label classification based on a predefined threshold (i.e., 0.5). To enable UFET to detect unknown-type instances, we classify instances whose probabilities of all types fall below the threshold as unknown types.

- Box4Types (Onoe et al., 2021) employs box embeddings to jointly represent entity mentions and entity types, thus effectively capturing their intersections. Based on these intersections, conditional probabilities for each type are computed to perform the final type prediction based on a predefined threshold (i.e., 0.5). Similarly, we regard instances whose conditional probabilities for all types fall below the threshold as unknown types.

- UniST (Huang et al., 2022) exploits type semantics to learn a joint semantic embedding space for both entity mentions and types. Depending on the similarities between mentions and types, types with similarities above a certain threshold are given as the final prediction, where the threshold is tuned on the validation set. Similar to the above, we regard instances with all similarities below this threshold as unknown types.

- CASENT (Feng et al., 2023) is a seq2seq model designed for entity typing that predicts types with calibrated confidence scores. To enable it the ability to detect unknown-type instances, we classify instances as unknown types if all of their type scores fall below a predefined global threshold, which is estimated on the validation set.

- ChatGPT (Ouyang et al., 2022) is a well-known generative large language model that has achieved success in many NLP applications. We

BBN	Few-NERD
Known Types in BBN: /person, /organization, /location, /gpe, /work_of_art, /product, /animal, /organization/corporation, /organization/educational, /organization/hotel, /organization/government, /organization/hospital, /organization/museum, /organization/political, /organization/religious, /location/continent, /location/lake_sea_ocean, /location/region, /location/river, /gpe/city, /gpe/country, /gpe/state_province, /work_of_art/book, /work_of_art/play, /work_of_art/song, /product/vehicle, /product/weapon	**Known Types in Few-NERD:** /person, /organization, /location, /building, /person/actor, /person/artist_author, /person/athlete, /person/director, /person/other, /person/politician, /person/scholar, person/soldier, /organization/company, /organization/education, organization/government, /organization/media_newspaper, /organization/other, /organization/political_party, /organization/religion, /organization/show_organization, /organization/sports_league, /organization/sports_team, /location/bodies_of_water, /location/gpe, /location/island, /location/mountain, /location/other, /location/park, /location/road_railway_highway_transit, /building/airport, /building/hospital, /building/hotel, /building/library, /building/other, /building/restaurant, /building/sports_facility, /building/theater
Unknown Types in BBN: /contact_info, /event, /facility, /disease, /game, /language, /law, /plant, /substance, /contact_info/url, /event/hurricane, /event/war, /facility/airport, /facility/attraction, /facility/bridge, /facility/building, /facility/highway_street, /substance/chemical, /substance/drug, /substance/food	**Unknown Types in Few-NERD:** /art, /event, /other, /product, /art/broadcast_program, /art/film, /art/music, /art/other, /art/painting, /art/written_art, /event/attack_battle_war_military_conflict, /event/disaster, /event/election, /event/other, /event/protest, /event/sports_event, /other/astronomy_thing, /other/award, /other/biology_thing, /other/chemical_thing, /other/currency, /other/disease, /other/educational_degree, /other/god, /other/language, /other/law, /other/living_thing, /other/medical, /product/airplane, /product/car, /product/food, /product/game, /product/other, /product/ship, /product/software, /product/train, /product/weapon

Table 5: Entity type composition for the two created datasets.

regard it as a baseline and evaluate its performance of addressing the OSET task with a tailor-designed prompt, which can be found in Figure 4.

- Llama3 (AI@Meta, 2024) is a large language model released by Meta, pre-trained on 15 trillion tokens. Similar to ChatGPT, we evaluate its performance for OSET with the same prompt in Figure 4.

- DeepSeek-V3 (Liu et al., 2024) is a powerful Mixture-of-Experts (MoE) language model with 671B total parameters. Similar to ChatGPT, we evaluate its performance on the OSET task using the same prompt shown in Figure 4.

(ii) *Open-set text classification.* As entity typing can be viewed as a text classification task to some extent, we also apply several open-set text classification methods to the OSET task for a comprehensive comparison. During inference of these methods, we convert the predicted type label into a type path via obtaining the corresponding types from the top level of the taxonomy to the level of the predicted type label.

You are an AI assistant who specializes entity types. Your task is as follows:
according to the sentence, predict the entity type of entity mention in the sentence. If the predicted type belongs to the known types supported by the system, return the corresponding known type, otherwise return "unknown". The supported known types include: {**known types**}. Only provide one type from above known types or "unknown" and do not give the explanation.

Figure 4: Prompt for LLM baselines.

- MSP (Hendrycks and Gimpel, 2017) is a softmax prediction probability baseline that classifies known-type instances based on the maximum softmax probabilities and detects those of unknown types with a predefined threshold (i.e., 0.5).

- SEG (Yan et al., 2020) utilizes a Gaussian mixture distribution to learn representations of instances and injects dynamic type semantic information into Gaussian means to enhance the detection of unknown-type instances.

- SCL (Zeng et al., 2021) is a supervised con-

trastive learning method, which learns discriminative representations for classifying known-type instances and detecting those of unknown types. Besides, it also adopts an adversarial augmentation mechanism to derive pseudo various views of known-type instances in the latent space, thus enhancing the performance.

- APRL (Chen et al., 2021) learns representations by maximizing variance between reciprocal points and known-type instances, and then leverages a learnable margin to constrain open space. However, directly applying this method from computer vision to our task significantly declines performance. Accordingly, we introduce a pre-training step using known-type instances and detect unknown-type instances with a probability threshold (i.e., 0.5).
- SELFSUP (Zhan et al., 2021) trains a discriminative classifier by constructing synthetic outliers via self-supervision, enhancing the capability of classifying known-type instances and detecting unknown-type instances.

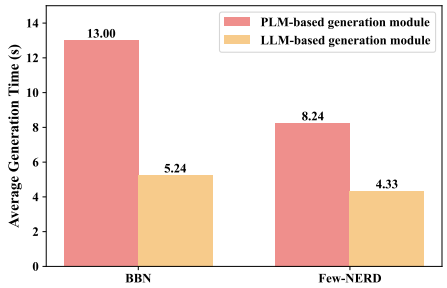


Figure 5: The average generation time of pseudo instances from PLM-based and LLM-based generation modules.

E Time Cost Analysis for Two-Stage Generation

We report the average generation time (in seconds per instance) of pseudo instances for PLM-based and LLM-based generation modules on BBN and Few-NERD datasets in Figure 5. As shown in this figure, the PLM-based generation module exhibits a significantly higher average generation time than the LLM-based module on both datasets. This may be attributed to the fact that the PLM-based generation module relies on identifying type-related keywords followed by multiple substitution iterations, whereas the LLM-based module directly synthesizes pseudo instances via in-context learning, thereby reducing computational overhead.

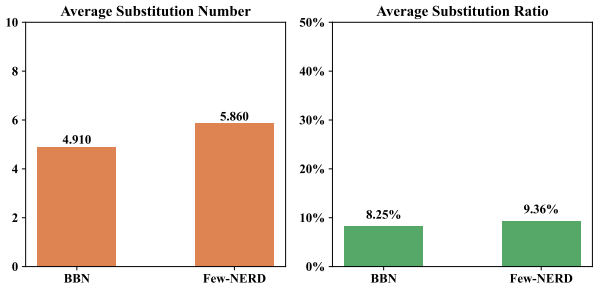


Figure 6: The average substitution number and average substitution rate of pseudo instances from the PLM-based generation module.

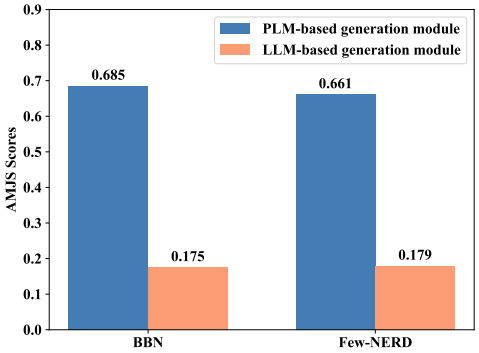


Figure 7: Comparing the diversity of pseudo instances from PLM-based and LLM-based generation modules.

F Case Study

To qualitatively analyze the effectiveness of our proposed two-stage generation model, we show the original known-type instances alongside pseudo unknown-type instances produced by the PLM-based (shown in the upper part of the table) and LLM-based (shown in the lower part of the table) generation modules in Table 6, where the entity mention within each instance is enclosed in square brackets. Moreover, in Table 6, we provide the labeled types of original instances obtained from the training set $\mathcal{M}$ and manually annotate the types for all pseudo unknown-type instances. In the upper part of Table 6, the original instances are drawn from the Few-NERD dataset, where the green tokens indicate substituted type-related keywords identified by the keyword extractor. Based on these original instances, the PLM-based generation module is leveraged to substitute their type-related keywords (green tokens) with appropriate candidate terms (red tokens) to create the corre-

Original & pseudo instances from PLM-based quality-targeted generation module		*Types*
Original instance	Kamada was a native of Ehime prefecture in [Shikoku island], Japan.	*known*: /location/island
Pseudo instance	Kamada was a native of Ehime prefecture in [Shikoku movie], Japan.	*unknown*: /art/film
Original instance	Garcia Combs taught writing and directing at [Cypress College] in southern CA.	*known*: /organization/education
Pseudo instance	Garcia Combs taught parody and comedy at [attending dances] in southern CA.	*unknown*: /art/other
Original instance	[Delmonico's restaurant] in New York City, which opened in 1827 and stayed open for almost 100 years, has been described as the most famous steak restaurant in American history.	*known*: /building/restaurant
Pseudo instance	[Delmonico's song] in New York City, which opened in 1827 and stayed open for almost 100 years, has been described as the most famous love song in American history.	*unknown*: /art/music
Pseudo instances from LLM-based diversity-targeted generation module		*Types*
The [BMW] is a high-performance hybrid car.		*unknown*: /product/car
The [Nobel Prize], an annual international award recognizing outstanding contributions in the fields of science, literature, and peace.		*unknown*: /other/award
The new policy aimed to reduce [carbon emissions] and promote subs.		*unknown*: /other/biology_thing
The Google AI lab is working on a new machine learning model for predicting [natural disasters].		*unknown*: /event/disaster
[Java] is a popular programming language developed by Sun Microsystems (now owned by Oracle Corporation).		*unknown*: /other/language
The celebrated novel, [The Lord of the Rings] was written by J.R.R. Tolkien.		*unknown*: /art/written_art

Table 6: Examples of pseudo instances generated by our two-stage generation model.

sponding pseudo unknown-type instances. We can notice that the types of original instances are transformed from known to unknown by only substituting several critical type-related keywords with suitable candidate terms. Additionally, it can be seen that these pseudo unknown-type instances (derived by the PLM-based generation module) preserve the maximum surface similarities with their corresponding original instances, thus ensuring their high quality. To quantitatively examine the keyword substitution of pseudo instances generated by the PLM-based module, we report the average substitution number and average substitution ratio of these pseudo instances on BBN and Few-NERD datasets in Figure 6. As illustrated in this figure, the average substitution ratio remains below 10% for both datasets, which confirms that high-quality pseudo instances can be successfully generated by substituting only a small proportion of keywords. Meanwhile, we notice that instances in Few-NERD require more substitution iterations than those in BBN, which can be attributed to the more implicit semantics of instances in Few-NERD. Despite the high generation quality of these pseudo instances, they inevitably hold almost identical entity context patterns with their corresponding original instances, resulting in limitations from the perspective of diversity. Fortunately, our proposed LLM-based generation module is capable of producing diverse pseudo unknown-type instances as shown in the lower part of Table 6. It can be observed that these pseudo instances produced by LLMs have a broad variety of topics and domains. To further quantitatively measure the diversity of generated pseudo instances by these two generation modules, we follow some data augmentation methods (Ghosh et al., 2023; Li et al., 2023) to calculate the average maximum Jaccard similarity (AMJS) between pseudo instances and all known-type instances, as presented in Figure 7. Notably, a lower AMJS score indicates a greater diversity of instances. We can find that the LLM-based generation module achieves significantly lower AMJS scores on both datasets compared with the PLM-based generation module. This finding confirms the superiority of this LLM-based generation module from the aspect of instance diversity, which fully exploits LLMs' rich implicit knowledge and remarkable generative capabilities to produce diverse pseudo unknown-type instances across different topics and domains.

G Related Work

G.1 Fine-Grained Entity Typing

Fine-grained entity typing (FET) aims to assign each entity mention a set of fine-grained types by providing its context and a predefined type taxonomy. One important line of FET methods mainly focuses on modeling the type hierarchy. For instance, HMGCN (Jin et al., 2019) develops a hierarchical multi-graph convolutional network to capture different semantic relatedness between entities. To naturally capture the hierarchical cor-

relations between entity types, Box4Types (Onoe et al., 2021) represents both entity mentions and entity types within a box embedding space that encodes label dependencies. Furthermore, to enhance the distinguishability between similar types, PICOT (Zuo et al., 2022) introduces a constrained contrastive strategy that directly models type distinctions. Another line of research has attempted to formulate the FET task into other tasks that facilitate more effective label semantics. Specifically, PLET (Ding et al., 2022) constructs entity-oriented prompts and formalizes FET as a cloze-style task, while CASENT (Feng et al., 2023) redefines FET as a sequence-to-sequence generation problem, enabling the prediction of entity types with calibrated confidence scores. By regarding FET as a semantic typing problem, UniST (Huang et al., 2022) proposes a unified framework to learn a joint semantic representation space such that candidate labels can be ranked based on their affinity with entity mentions.

However, these FET methods operate under the closed-set assumption, lacking the capability to detect unknown-type instances. Additionally, zero-shot FET (Yuan and Downey, 2018; Zhang et al., 2020a, 2024b) has been investigated to alleviate the human annotation burden by generalizing from seen types to unseen ones. For instance, OTyper (Yuan and Downey, 2018) maps mention embeddings to a type embedding space by training a neural model that integrates both entity and contextual information. To further capture the relationship between seen and unseen types, MZET (Zhang et al., 2020a) employs a memory network to incorporate the hierarchical structure into the entity type representation. Moreover, SETYPE (Zhang et al., 2024b) proposes to study seed-guided fine-grained entity typing in science and engineering domains, which aims to perform zero-shot FET with weaker signals. Nonetheless, these zero-shot FET methods still require unseen type names as the necessary input during inference, which inherently holds the closed-set assumption and cannot be applied to address the OSET task.

G.2 Open-Set Text Classification

Open-set text classification aims to classify textual instances into known categories while simultaneously detecting instances that belong to unknown categories that emerge during testing. Early methods (Jain et al., 2014; Rifkin and Klautau, 2004) leverage traditional machine learning methods (e.g., SVM) to detect instances of unknown categories. However, these methods highly rely on feature engineering, which is labor-intensive. With the advance of deep learning, researchers employ deep neural networks for open-set text classification. For example, DOC (Shu et al., 2017) employs a one-vs-all classifier to compute its maximum probability for detecting instances of unknown categories based on a predefined threshold. SEG (Yan et al., 2020) employs a Gaussian mixture distribution for representations and injects dynamic class semantic information into Gaussian means to improve the detection of unknown categories. Besides, SCL (Zeng et al., 2021) proposes a supervised contrastive learning method to increase intra-class coherence and inter-class variation, thus capturing discriminative representations for the detection of unknown categories. To alleviate the shortage of unknown-category training instances, SELFSUP (Zhan et al., 2021) constructs pseudo instances to train a discriminative classifier, thereby enhancing the capability of classifying known-category instances and detecting those of unknown categories. However, these open-set text classification methods inherently neglect the hierarchical structure of the type taxonomy, resulting in their inability to effectively address the OSET task, as shown in our experiments.