Leveraging Label Semantics and Entity Description Generation for LLM-based Fine-grained Entity Typing

Anonymous ACL submission

Abstract

Fine-grained entity typing (FET) aims to assign semantically rich and contextually appropriate types to entity mentions. While recent studies have explored the use of large language models (LLMs) for this task, two key challenges persist. First, FET typically involves a large number of entity types, making it difficult for LLMs to perform accurate classification. Second, the presence of label noise in the training data introduced by automatic supervision methods hinders effective fine-tuning. To address these challenges, we propose **DR-FET**, a descriptor-based retrieval-augmented framework that reduces the effective label space and constructs high-precision training data under noisy supervision. Our method introduces natural language descriptors as an intermediate semantic representation for both entity mentions and types. During inference, entity descriptors are used to retrieve a small set of semantically relevant candidate types, enabling the LLM to perform fine-grained classification under explicit candidate constraints. During training, the same descriptor and retrieval mechanism is reused to identify high-confidence instances from distantly supervised data, prioritizing label precision for efficient fine-tuning. Experiments on two widely used benchmarks demonstrate that the proposed method consistently outperforms existing fine-grained entity typing approaches under noisy supervision.

1 Introduction

Fine-grained entity typing (FET) aims to assign precise and context-sensitive types to entity mentions. Despite its importance for many downstream tasks, FET remains challenging due to the large label space and the lack of high-quality annotated data.

Large language models (LLMs) have shown strong potential for FET by capturing rich contextual semantics. However, directly applying LLMs to fine-grained typing is problematic. First, typical FET benchmarks contain dozens or even hundreds of types, and prompting an LLM to select from the full ontology often leads to unreliable predictions.Our preliminary analysis indicates that a strong LLM achieves 65.61% strict accuracy when choosing from five candidate types, but performance drops to 29.4% when 50 types are considered (Appendix A). Second, due to the high cost of manual annotation, existing approaches commonly rely on distant or weak supervision, which introduces substantial label noise and further complicates learning over large type sets.

To address these challenges, we propose **DR-FET**, a descriptor-based retrieval-augmented framework for fine-grained entity typing under distant supervision. The key idea is to introduce natural language descriptors as an intermediate semantic representation. Given an entity mention and its context, an LLM generates concise descriptors that capture the entity's semantic role. These descriptors are then matched with textual type representations using a frozen text encoder to retrieve a compact candidate set with high recall. The final prediction is made by an LLM constrained to select only from this candidate set, transforming fine-grained typing into a structured decision problem.

Moreover, the same descriptor-based retrieval mechanism is leveraged during training to construct a high-confidence dataset from noisy distant supervision. By retaining only instances where retrieved candidates agree with distant labels, we obtain a compact but reliable training set that enables effective LLM fine-tuning without additional annotation.

Our contributions are summarized as follows:

- We propose a descriptor-based retrieval-augmented framework that reduces large type spaces into compact candidate sets for reliable

LLM-based fine-grained entity typing.

- We introduce a precision-first training data construction strategy that extracts high-confidence supervision from noisy distant labels using the same retrieval mechanism as inference.

- Extensive experiments on benchmark datasets demonstrate the effectiveness and robustness of the proposed approach.

2 Related Work

Fine-grained Entity Typing (FET) has been widely studied under different supervision settings. We review prior work from three perspectives: noise-aware learning under distant supervision, model-centric approaches to FET, and recent advances leveraging retrieval and large language models.

2.1 Noise-aware Fine-grained Entity Typing

Distant supervision is a common strategy for constructing FET training data, but it inevitably introduces label noise, including false positives, missing labels, and confirmation bias. Early work such as NFETC-AR (Onoe and Durrett, 2019) mitigates noise by modeling latent clusters of entity mentions and propagating labels within similarity graphs. Subsequent studies explore explicit noise handling through probabilistic relabeling (Zhang et al., 2021), cluster-level loss correction (Pang et al., 2022), and posterior-based label denoising without external supervision (Pan et al., 2022). These approaches improve robustness by explicitly modeling or correcting noisy labels during training.

2.2 Model-centric and LLM-based FET

Beyond noise modeling, many methods focus on improving entity typing through better representation learning or semantic reasoning. FiveFine (Dai and Zeng, 2023) generates context-dependent hypernyms using masked language models to support fine-grained and ultra-fine typing with minimal annotation. More recent work explores the use of large language models and structured semantic priors. OntoType (Komarlu et al., 2024) integrates ontology hierarchies with prompting and natural language inference to refine candidate type selection. EnCore (Mtumbuka and Schockaert, 2024) enhances entity representations through entity-aware attention and entity-centric pretraining objectives.

While these methods demonstrate strong performance, they typically assume relatively reliable supervision or operate directly over the full label space, which limits their robustness under large ontologies and noisy distant supervision.

2.3 LLM for Information Extraction

In parallel, large language models have shown strong potential in information extraction tasks, including entity and relation extraction (Xu et al., 2024). Recent work such as Co-Prediction (Tang et al., 2023) improves robustness under noisy supervision by exploiting disagreement among multiple predictions. Different from these approaches, our method incorporates LLMs through descriptor generation and candidate-constrained inference, enabling scalable and robust fine-grained entity typing without explicit noise modeling.

2.4 Retrieval and LLMs for Information Extraction

Embedding-based retrieval has been successfully applied in large-scale recommendation (Cheng et al., 2016), semantic search (Huang et al., 2020), and document retrieval (Zeng et al., 2022), but remains underexplored in fine-grained entity typing, where most methods score all labels exhaustively. Inspired by retrieval-classify paradigms, we introduce descriptor-based semantic retrieval to narrow the candidate label space before final prediction.

3 Methodology

We propose a descriptor-based framework for fine-grained entity typing that explicitly seperates semantic representation, label space reduction, and final decision making. Our inference procedure follows a three-stage pipeline:

1. **Descriptor-based Semantic Abstraction**, which converts entity mentions into natural language descriptors capturing their semantic roles;

2. **Semantic Candidate Type Retrieval**, which reduces the large type space to a small set of relevant candidate types using semantic similarity;

3. **Candidate-constrained Type Inference with LLM**, which performs the final type prediction within the retrieved candidate set.

In addition to the inference pipeline, we design a training data construction strategy tailored for distantly supervised data. This strategy leverages the same descriptor generation and retrieval components to identify high-confidence training instances,

enabling effective fine-tuning of large language models under noisy supervision.

3.1 Descriptor-based Semantic Abstraction

(1) Entity Descriptor Generation. We adopt a few-shot prompting strategy to guide the model to produce coherent and semantically relevant descriptive phrases (descriptors). The complete prompt template is provided in Appendix B.2. For each entity mention, we prompt the LLM to generate 3–5 descriptors that characterize the entity's semantic role. For the training set, gold labels are included in the prompt to encourage the generation of descriptors that are well-aligned with the ground truth types. This enhances label-descriptor consistency but may reduce the model's generalization capability. For the validation and test sets, gold labels are omitted to better simulate real-world scenarios where true types are unknown. In these cases, the model generates more diverse yet contextually appropriate descriptors.

We note that descriptor generation follows different supervision settings across training stages. During training data construction under distant supervision in Section 3.4, gold labels are not provided to the LLM, as distant annotations are inherently noisy and may be underspecified. In contrast, during the final model fine-tuning stage, descriptor generation is conditioned on gold labels, as the filtered training set is already high-precision, and label-aware prompting helps improve descriptor–type alignment.

(2) Label Descriptor Aggregation. For each label in the entity type set, we aggregate descriptors generated from training instances annotated with that label. Specifically, we retain the top-K most frequent descriptor phrases as representative summaries of the label semantics. These label descriptors are later used to compute semantic similarity with entity descriptors during candidate retrieval.

3.2 Semantic Candidate Type Retrieval

After obtaining a set of natural language descriptors for each label and the given entity, our goal is to identify a small subset of relevant labels from the full label space. This step performs candidate narrowing to reduce the computational and semantic burden on the final classification stage.

Embedding Strategy. As shown in Stage 2 of Figure 1, we project each label or entity descriptor into a shared semantic space using a pretrained text encoder. As a result, for each label or entity, the multiple descriptors constructed in the previous step result in a corresponding set of embeddings for it.

Candidate Selection. For each descriptor $d_e^{(i)}$ of an entity e, we retrieve its top-q most similar label-descriptor pairs based on cosine similarity:

$$\mathcal{P}_q^{(i)} = \text{Top-}q \left\{ \left(l, \cos\left(\mathbf{v}_{d_e^{(i)}}, \mathbf{v}_{d_l} \right) \right) \,\middle|\, l \in \mathcal{L},\ d_l \in D_l \right\},$$

where $\mathbf{v}_{d_e^{(i)}}$ and $\mathbf{v}_{d_l}$ denote the embedding vectors of the entity descriptor $d_e^{(i)}$ and the label descriptor d_l, respectively; $\mathcal{L}$ is the full label set, and D_l is the set of descriptors associated with label l.

For each entity, we compute a label score by summing over similarities between its descriptors and the retrieved label-descriptor pairs:

$$\text{Score}(e, l) = \sum_{i=1}^{m} \sum_{(l^*, d_{l^*}) \in \mathcal{P}_q^{(i)}} \mathbf{1}_{\{l^* = l\}} \cdot \cos\left(\mathbf{v}_{d_e^{(i)}}, \mathbf{v}_{d_{l^*}} \right).$$

Here, $\mathbf{1}_{\{l^*=l\}}$ is the indicator function that equals 1 if $l^* = l$ and 0 otherwise. Intuitively, this score reflects the accumulated similarity between descriptors of entity e and those associated with label l among the top-q most similar pairs.

Finally, we select the top-k scoring labels as candidate types for entity e:

$$\mathcal{Y}_{\text{cand}}(e) = \text{Top-}k_{l \in \mathcal{L}} \ \text{Score}(e, l).$$

The parameter q controls descriptor-level recall during retrieval, while k bounds the candidate set size for final inference, balancing recall and decision complexity. The complete procedure is detailed in Appendix C.

3.3 Candidate-constrained Type Inference with LLM

For final prediction, we fine-tune an open-source LLM using LoRA on the filtered high-confidence examples. Given an input text, entity mention, and its candidate labels with optional descriptors, the model predicts all applicable labels in a multi-label format. We adopt a zero-shot-style prompting setup during both training and inference, and apply LoRA to reduce GPU memory usage while maintaining flexibility. All prompt templates are provided in Appendix B.3.

3.4 Training Data Construction

Fine-grained entity typing under distant supervision is often affected by noisy and underspecified annotations, where entity mentions are labeled with coarse-grained types that only partially reflect their

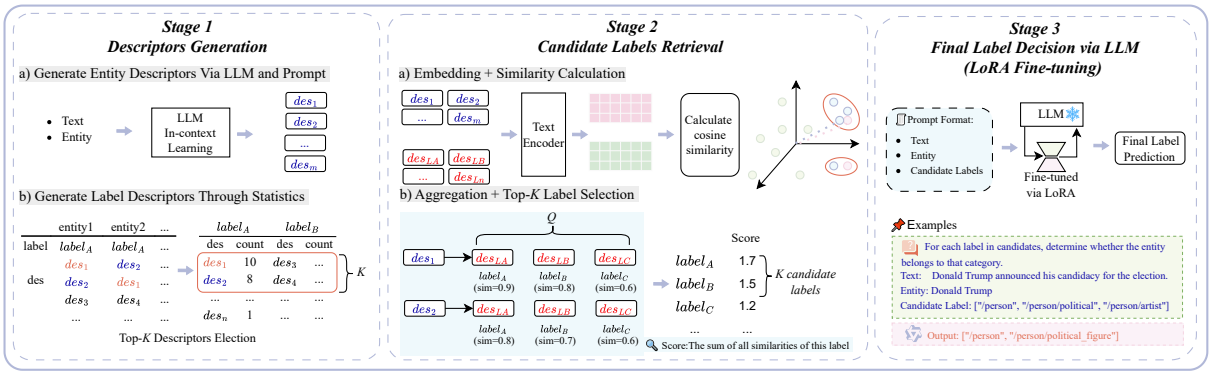

Figure 1: Overview of the proposed descriptor-based retrieval-augmented framework for fine-grained entity typing. The pipeline consists of three stages: (1) descriptor generation, where an LLM produces natural language descriptions for an entity mention given its context; (2) semantic candidate type retrieval, where both entity and label descriptors are embedded and matched to retrieve a compact set of candidate types; and (3) candidate-constrained LLM inference, which predicts the final entity types from the retrieved candidates.

true semantics. Directly using such labels as hard supervision can bias the model toward overly conservative predictions and hinder fine-grained discrimination. To address this issue, we construct a high-confidence training set through a two-stage consistency-based instance selection strategy that explicitly leverages the hierarchical structure of the type ontology. An overview of this process is shown in Figure 2.

3.4.1 Syntactic Consistency Verification

As an initial denoising step, we apply syntactic consistency verification to remove invalid entity mentions produced by distant supervision. Specifically, we verify whether each mention corresponds to a valid nominal phrase in context, filtering out malformed spans and non-entity expressions. This step is performed using lightweight linguistic checks and does not involve semantic interpretation. Only instances that pass this syntactic verification stage are retained for subsequent processing.

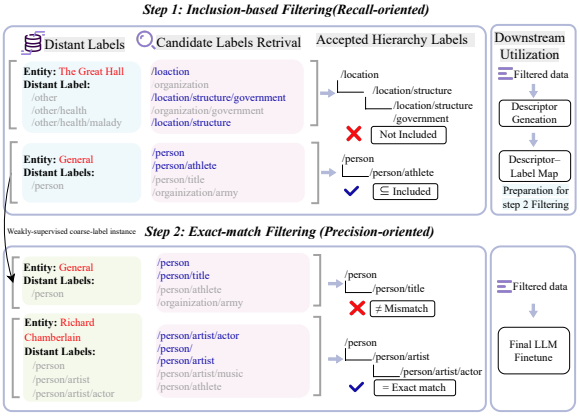

Figure 2: Two-stage consistency-based training data construction under distant supervision. Given noisy and potentially underspecified distant labels, we first apply an inclusion-based consistency check to retain recall-oriented instances whose distant labels are hierarchically covered by at least one accepted candidate label chain. The retained instances are then re-evaluated using a strict exact-match criterion to remove coarse-label-only and partially correct annotations, yielding a high-precision training set for LLM fine-tuning.

3.4.2 Consistency-based Instance Selection

For each training instance that passes syntactic verification, we reuse the descriptor-based semantic abstraction (Section 3.1 and candidate type retrieval (Section 3.2) procedures to obtain an ordered list of candidate types. Instead of evaluating individual labels in isolation, we construct acceptance chains from the retrieved candidates. An acceptance chain is defined as a hierarchically valid sequence of types in which each successive type is a descendant of the previous one, forming a coherent semantic refinement path in the ontology.

We consider an instance to be semantically consistent if there exists at least one acceptance chain whose completed hierarchy covers all distant supervision labels associated with the instance. This criterion ensures that the retrieved candidate types provide a plausible hierarchical explanation for the distant annotation, even when the annotation itself is coarse-grained.

However, semantic consistency alone does not guarantee reliable fine-grained supervision. Therefore, we further apply a strict matching criterion to select high-confidence training instances. Specifically, an instance is retained for fine-tuning only if

the accepted hierarchical chain exactly matches the distant supervision labels after hierarchy completion. This second-stage filtering yields a subset of instances whose labels are both semantically consistent and sufficiently specific to serve as reliable fine-grained supervision.

Importantly, this instance selection process is applied only during training data construction and does not affect the inference pipeline. By separating semantic consistency verification from supervision reliability enforcement, our method effectively mitigates the impact of underspecified distant labels while preserving high-quality training signals for fine-grained type prediction.

4 Experiments

4.1 Datasets

Datasets	OntoNotes	BBN
Types	89	47
Hierarchy Depth	3	2
Mentions (Train)	253,241	86,078
Mentions (Step 1)	58,400	25,800
Mentions (Step 2)	29,800	12,024
Mentions (Step 2 Sampled)	9,806	6,120
Mentions (Test)	8,963	13,531
One-label Test (%)	94	100

Table 1: Statistics of the fine-grained entity typing datasets used in our experiments.

We evaluate our approach on two widely used public datasets for fine-grained entity typing. **OntoNotes** (Weischedel et al., 2013) consists of sentences from the OntoNotes corpus that are distantly annotated using DBpedia Spotlight (Gillick et al., 2016), resulting in 253K training mentions across 89 entity types. **BBN** (Weischedel and Brunstein, 2005) contains sentences extracted from the Wall Street Journal and is distantly labeled using the same annotation pipeline, covering 47 entity types.

Following prior work on denoising distantly supervised FET data, we apply a two-stage filtering process to construct reliable training sets. Step 1 corresponds to syntactic consistency verification (Section 3.4.1), which removes malformed spans and non-entity mentions from the distantly supervised data. Step 2 applies the proposed consistency-based instance selection (Section 3.4.2), retaining only semantically consistent instances whose labels are hierarchically complete and suitable for fine-grained supervision. To mitigate label imbalance introduced by distant supervision, we further construct a balanced subset from Step 2 by sampling instances across entity types, this subset is denoted as *Step 2 Sampled* in Table 1.

Compared to early experimental variants that relied on a small subset of the distantly supervised data, our final experimental setup leverages the full distant supervision corpus and relies on the proposed filtering strategy to ensure training data quality. Detailed dataset statistics after each filtering stage are reported in Table 1.

4.2 Metrics and Implementation Details

We follow prior work and evaluate model performance using strict accuracy (Acc), Macro-F1 (Ma-F1), and Micro-F1 (Mi-F1). All reported results are averaged over five runs with different random seeds to reduce variance introduced by data sampling and LoRA initialization.

We employ the open-source **Qwen2.5-7B-Instruct** (Qwen Team, 2024) model as the core large language model throughout the framework. The LLM is used for both entity descriptor generation and final candidate-constrained type inference, with LoRA applied for parameter-efficient fine-tuning. For semantic candidate retrieval, we use **BERT** (Devlin et al., 2018) as a frozen text encoder without updating its parameters to embed entity and label descriptors into a shared semantic space. In addition to the main model, we also report results using other open-source LLMs for comparison (Table 5).

Details of hyperparameters, prompt templates, and training configurations are summarized in Appendix D.

4.3 Baseline Methods

To evaluate the effectiveness of our approach, we compare it with a set of representative baseline methods for fine-grained entity typing. These baselines are selected to reflect different strategies for handling large label spaces and noisy supervision under distant labeling. Following their core methodological characteristics, we group them into three categories.

Distantly Supervised and Noise-aware FET Models. This category includes methods that explicitly address label noise and confirmation bias introduced by distant supervision, and represents the dominant paradigm in fine-grained entity typing. We consider the following representative

approaches: **NFETC-AR** and **NFETC$_{\text{hier}}$** (Onoe and Durrett, 2019), which incorporate partial-label learning and hierarchical constraints; **NFETC-VAT** (Shi et al., 2020) and **CLSC-VAT** (Chen et al., 2019), which apply consistency regularization and virtual adversarial training to improve robustness; **FCLC** and **FCLC$_{\text{hier}}$** (Pang et al., 2022), which divide noisy supervision into fine-grained candidate sets; **DenoiseFET** (Pan et al., 2022), which performs automatic noise detection and correction; **Co-Prediction** (Tang et al., 2023), which leverages mutual supervision between models; and **FiveFine** (Dai and Zeng, 2023), which gradually refines coarse labels into fine-grained predictions. All these methods are based on encoder models trained under distantly supervised settings and serve as strong and widely adopted baselines.

Ontology-driven and Hierarchy-aware Typing. Methods in this category exploit type ontologies and hierarchical structures to guide fine-grained entity typing, without explicitly modeling instance-level label noise. We include **OntoType** (Komarlu et al., 2024), which utilizes ontology structures and type descriptions to improve semantic alignment between mentions and labels.

Representation Enhancement with Additional Pre-training. This category includes methods that improve entity typing performance by enhancing representation learning through additional pre-training objectives. We include **EnCore** (Mtumbuka and Schockaert, 2024), which retrains the BERT encoder using entity-aware attention and specialized objectives on cleaned supervision. Since EnCore does not operate on the original distantly supervised data and adopts a different training protocol, its results are not strictly comparable to other baselines under our setting, and we report the metrics provided in the original paper.

In addition, for **FiveFine** and **FCLC**, we report results obtained by retraining the original models on the high-confidence training data constructed by our framework. This setting isolates the effect of improved supervision quality without modifying the original model architectures, and demonstrates the general applicability of our data filtering strategy.

4.4 Main Results and Analysis

Some baseline methods are not included in the BBN evaluation. This is because the original papers did not report results on this dataset, and the corresponding implementations are either unavailable or rely on proprietary preprocessing pipelines, making reliable reproduction infeasible. EnCore retrains the BERT encoder with specialized objectives and does not operate on the original distantly supervised data used by other baselines. As a result, strict accuracy is not reported in the original work and is not directly comparable under our evaluation setting

Model	Strict Acc	Macro F1	Micro F1
NFETC-AR	62.8	77.8	71.8
w / *hier*	64.0	78.8	73.0
FCLC$_{\text{hier}}$	65.3	79.6	74.0
DenoiseFET	59.2	81.3	75.2
Co-Prediction	64.6	87.1	81.7
FiveFine	63.7	86.2	80.6
OntoType	65.7	81.5	73.4
EnCore	–	88.9	83.4
DR-FET	**73.7**	**89.3**	**85.2**

Table 2: FET performance on OntoNotes. Our method is listed in the last row.

Model	Strict Acc	Macro F1	Micro F1
NFETC-VAT	76.7	80.7	80.9
NFETC-AR	76.7	81.4	81.5
CLSC-VAT	76.9	81.2	81.4
FCLC	82.0	86.2	86.7
FiveFine	67.2	86.3	85.2
FCLC$_{\text{hier}}$	79.0	84.2	84.8
DR-FET	**83.1**	**88.3**	**86.4**

Table 3: FET performance on BBN. Our method is shown in the last row.

Tables 2 and 3 summarize the performance of our method and representative prior approaches on the OntoNotes and BBN datasets. Both on OntoNotes and BBN, DR-FET achieves the best performance across all evaluation metrics. In particular, it substantially improves strict accuracy over previous methods, indicating that explicitly constraining the prediction space with semantically retrieved candidates enables more reliable selection of exact fine-grained types. Gains in both Macro F1 and Micro F1 further suggest that the proposed method performs consistently well across both head and tail types. We also observe that retraining existing noise-aware models such as FiveFine and FCLC on the filtered high-confidence training data leads to notable F1 improvements, suggesting that the proposed data construction strategy is model-agnostic and can benefit a wide range of FET approaches.

Overall, these results demonstrate that the pro-

posed descriptor-based retrieval and candidate-constrained inference framework effectively addresses both large label spaces and noisy distant supervision in fine-grained entity typing.

5 Ablation

All additional analyses are conducted exclusively on the OntoNotes dataset. We choose OntoNotes for these experiments because it contains a large and diverse set of entity types, providing a representative testbed for evaluating the effectiveness and robustness of our DR-FET pipeline. Conducting these analyses on a single dataset allows us to present detailed results and visualizations while avoiding excessive redundancy across datasets, without affecting the general conclusions regarding the contribution of each component.

5.1 Impact of Training Strategy

Method	Strict Acc	Macro F1	Micro F1
ICL	24.4	55.9	56.3
FT-Raw	20.3	63.5	60.5
FT-Filtered	56.9	70.9	68.3
ICL-Candidate	52.13	76.72	82.77
DR-FET	**73.7**	**89.3**	**85.2**

Table 4: Performance of different training strategies on OntoNotes.

We compare different training strategies using the same LLM backbone:

- **ICL**: direct in-context learning without any fine-tuning, no descriptors or candidate constraints.
- **FT-Raw**: fine-tuning on 10k randomly sampled distant supervision data without filtering.
- **FT-Filtered**: fine-tuning on the data constructed by our filtering method in Section 3.4.
- **DR-FET**: our proposed pipeline using descriptor-based semantic abstraction, retrieval-based candidate constraints, and high-confidence filtered training data.
- **ICL-Candidate**: in-context learning with retrieval-based candidate constraints, without fine-tuning or training data filtering.

All ablation studies are conducted on the OntoNotes dataset, which contains a large and diverse set of entity types and serves as a representative benchmark for analyzing the contribution of individual components in our framework.

Adding candidate constraints to ICL (ICL-Candidate) results in modest gains over standard ICL, suggesting that retrieval-based candidate restriction alone provides limited gains, and that its effectiveness is substantially amplified when combined with high-quality supervision through fine-tuning. Finally, our full DR-FET pipeline, which combines descriptor-based semantic abstraction, retrieval-based candidate constraints, and filtered high-confidence training data, achieves the best performance across all metrics. These results indicate that performance gains are primarily driven by the combination of high-precision training data and candidate-constrained inference, while descriptor-based abstraction further enables effective integration of the two.

5.2 Effect of Encoder and Candidate Set Size on Recall

To evaluate the effectiveness and robustness of the descriptor-based candidate retrieval stage (Stage 2), we measure the top-K recall of retrieved candidate labels using different sentence embedding models. We compare three pretrained encoders—**BERT** (Devlin et al., 2018), **BGE** (Xiao et al., 2023), and **E5** (Wang et al., 2022). As a comparison, we include a classification-based baseline (**BERT+Cls**), which ranks labels using logits from a BERT classifier trained on the full dataset.

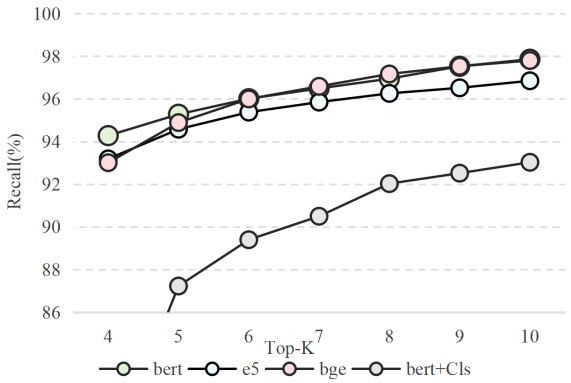

Figure 3: Top-K recall on test set for different embedding models with $Q = 5$ descriptors.

We empirically observe that varying the number of label descriptors has minimal impact on candidate recall, and therefore fix $Q = 5$ in all experiments. We vary the number of retrieved candidate labels K from 3 to 10 and compute $Recall@K$ as the proportion of entity mentions for which at least one gold label appears in the top-K retrieved candidates.

As shown in Figure 3, descriptor-based retrieval methods achieve high recall with a small number of candidates, exceeding 95% recall with fewer than seven labels. Among them, BERT performs best in the low-K regime, while E5 and BGE exhibit comparable performance as K increases, indicating that the retrieval framework is robust to the choice of embedding model.

In contrast, the classification-based baseline (BERT+Cls) performs substantially worse, particularly for small candidate sets. Even with $K = 10$, its recall remains noticeably lower than descriptor-based retrieval methods, demonstrating that directly ranking labels by classification logits is insufficient for high-recall candidate generation.

Overall, these results confirm that descriptor-based retrieval can generate compact candidate sets with high recall, enabling reliable candidate-constrained inference, while the generated descriptors are used only for retrieval and data construction and are not exposed to the final decision model, avoiding label information leakage.

5.3 Generalization Across LLMs

LLM	Strict Acc	Micro F1	Macro F1
Qwen2.5-3B-Instruct	72.19	**84.65**	**89.29**
Gemma-2-2B-it	**72.65**	84.47	88.06
LLaMA-2-7B-chat-hf	69.75	83.19	87.05

Table 5: Performance of our method with different LLM backbones.

To examine the generalizability of our method across different large language models, we evaluate it using several alternative open-source LLMs, including *Qwen2.5-3B-Instruct* (Qwen Team, 2024), *Gemma-2-2B-it* (Gemma Team, 2024), and *LLaMA=2-7B-chat-hf* (Touvron et al., 2023). These models differ in architecture and training recipes, providing a representative comparison across LLM families. As shown in Table 5, our framework achieves consistently strong performance across all models, with strict accuracy ranging from 69.7% to 72.6%, indicating that the proposed approach does not rely on a specific LLM backbone.

6 Conclusion

We study fine-grained entity typing under distant supervision, where large label spaces and noisy annotations pose significant challenges. To address these issues, we propose **DR-FET**, a descriptor-based retrieval-augmented framework that reframes entity typing as a candidate-constrained decision problem.

DR-FET leverages natural language descriptors to establish semantic alignment between entity mentions and type labels. A unified retrieval mechanism is used both to generate compact, high-recall candidate sets for inference and to construct high-confidence training data, enabling effective fine-tuning under noisy supervision.

Experimental results demonstrate that DR-FET consistently outperforms prior methods across benchmark datasets. Further analyses show that the proposed framework is robust to encoder choices and LLM backbones, and benefits substantially from precision-oriented data construction.

Future work includes exploring richer descriptor generation, adaptive candidate selection strategies, and more efficient retrieval schemes. Beyond entity typing, the proposed framework may generalize to other large-label classification tasks under weak or noisy supervision.

7 Limitations

First, the quality of the generated descriptors depends on the underlying LLM and the prompting strategy. In this work, descriptor generation relies on general-purpose in-context learning without task-specific prompt optimization, which may introduce variability in descriptor content. Designing more structured or controllable descriptor generation remains an open direction.

Second, candidate generation is based on approximate top-K semantic retrieval and does not guarantee perfect recall in all cases. Although high recall is achieved with a small candidate set in our experiments, rare or highly ambiguous entity types may still be missed.

Finally, DR-FET adopts a multi-stage pipeline that introduces additional computational cost compared to single-stage classifiers. Future work may explore adaptive routing strategies, such as easy-hard instance separation or dynamic candidate sizing, to improve efficiency without sacrificing performance.

References

Bo Chen, Xiaotao Gu, Yufeng Hu, Siliang Tang, Guoping Hu, Yueting Zhuang, and Xiang Ren. 2019. Im-

proving distantly-supervised entity typing with compact latent space clustering. *arXiv preprint*.

Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, pages 7–10, Boston, MA, USA. Association for Computing Machinery.

Hongliang Dai and Ziqian Zeng. 2023. From ultra-fine to fine: Fine-tuning ultra-fine entity typing models to fine-grained. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2259–2270, Toronto, Canada. Association for Cmputational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint*.

Gemma Team. 2024. Gemma. Accessed: 2024-12-26.

Dan Gillick, Nevena Lazic, Kuzman Ganchev, Jesse Kirchner, and David Huynh. 2016. Context-dependent fine-grained entity type tagging. *arXiv preprint*.

Jui-Ting Huang, Ashish Sharma, Shuying Sun, Li Xia, David Zhang, Philip Pronin, Janani Padmanabhan, Giuseppe Ottaviano, and Linjun Yang. 2020. Embedding-based retrieval in facebook search. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2553–2561, Virtual Event, CA, USA. Association for Computing Machinery.

Tanay Komarlu, Minhao Jiang, Xuan Wang, and Jiawei Han. 2024. OntoType: Ontology-guided and pre-trained language model assisted fine-grained entity typing. *arXiv preprint*.

Frank Mtumbuka and Steven Schockaert. 2024. EnCore: Fine-grained entity typing by pre-training entity encoders on coreference chains. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1768–1781, St. Julian's, Malta. Association for Computational Linguistics.

Yasumasa Onoe and Greg Durrett. 2019. Learning to denoise distantly-labeled data for entity typing. *arXiv preprint*.

Weiran Pan, Wei Wei, and Feida Zhu. 2022. Automatic noisy label correction for fine-grained entity typing. *arXiv preprint*.

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 *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1997–2006, Dublin, Ireland. Association for Computational Linguistics.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Haochen Shi, Siliang Tang, Xiaotao Gu, Bo Chen, Zhigang Chen, Jian Shao, and Xiang Ren. 2020. Alleviate dataset shift problem in fine-grained entity typing with virtual adversarial training. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*, pages 3898–3904, Yokohama, Japan. International Joint Conferences on Artificial Intelligence Organization.

Minghao Tang, Yongquan He, Yongxiu Xu, Hongbo Xu, Wenyuan Zhang, and Yang Lin. 2023. Learning to correct noisy labels for fine-grained entity typing via co-prediction prompt tuning. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 5848–5858, Singapore. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint*.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint*.

Ralph Weischedel and Ada Brunstein. 2005. BBN pronoun coreference and entity type corpus. Web Download. LDC Catalog No. LDC2005T33.

Ralph Weischedel, Sameer Pradhan, Lance Ramshaw, Martha Palmer, Nianwen Xue, Mitchell Marcus, John Conroy, Mona Diab, Olga Babko-Malaya, and Justin MacBride. 2013. Ontonotes: A large training corpus for enhanced processing. In Nancy Ide and James Pustejovsky, editors, *Handbook of Linguistic Annotation*, pages 93–134. Springer.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-Pack: Packaged resources to advance general chinese embedding. *Preprint*, arXiv:2309.07597.

Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, Yang Wang, and Enhong Chen. 2024. Large language models for generative information extraction: A survey. *arXiv preprint*.

Chaoliang Zeng, Layong Luo, Qingsong Ning, Yaodong Han, Yuhang Jiang, Ding Tang, Zilong Wang, Kai Chen, and Chuanxiong Guo. 2022. FAERY: An

FPGA-accelerated embedding-based retrieval system. In *Proceedings of the 16th USENIX Symposium on Operating Systems Design and Implementation*, pages 841–856, Carlsbad, CA. USENIX Association.

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 *Proceedings of the 29th International Joint Conference on Artificial Intelligence*, pages 3808–3815.

A Effect of Candidate Set Size

A.1 Effect of Candidate Set Size on LLM Performance

To quantify the effect of candidate label set size on LLM performance, we conduct a controlled experiment using in-context learning (ICL). For each entity, a candidate set of size N was randomly sampled from the full type ontology, always including the correct type, with N varying from 5 to 50. Table 6 reports the results, showing a consistent decline in classification accuracy as the number of candidates increases. These findings highlight that larger candidate sets introduce additional uncertainty for the LLM, thereby motivating the use of a more focused candidate-constrained inference strategy in our framework.

#Types	Acc (%)	Micro-F1	Macro-F1
5	64.31	81.92	87.44
10	52.13	76.72	82.77
15	47.32	73.91	79.41
20	42.42	70.28	75.39
30	40.17	68.97	74.86
40	35.64	65.31	67.28
50	29.41	61.43	59.54

Table 6: Performance with different candidate set sizes.

A.2 Effect of Encoder and Candidate Set Size on Recall

Top-1 – Top-5 Recall (%)					
Model	1	2	3	4	5
BERT	79.75	89.20	92.35	94.28	95.28
E5	77.30	88.18	90.98	93.19	94.58
BGE	73.40	85.56	90.90	93.01	94.89
BERT+Cls	40.00	60.00	70.00	79.00	87.23
Top-6 – Top-10 Recall (%)					
Model	6	7	8	9	10
BERT	96.01	96.49	96.95	97.53	97.86
E5	95.38	95.85	96.26	96.52	96.85
BGE	96.00	96.59	97.17	97.53	97.80
BERT+Cls	89.40	90.50	92.03	92.52	93.03

Table 7: Top-k recall (%) of different encoders for candidate type generation on OntoNotes.

B Prompt Templates and In-Context Examples

B.1 Syntactic Consistency Verification Prompt Template

You are a helpful assistant.
Task* You are a professional text analysis assistant. Please judge whether the phrase is valid data according to the following requirements:
1. determine which sentence component the input phrase is
2. invalid input: preposition, pronoun, conjunction, verb, adjective, adverb, incomplete sentence
3. valid input: named entity, general noun, abstract noun
-Example:
✓dogs → valid (general noun)
✓Beijing → valid (named entity)
✗ he → not valid (pronoun)
✗ On → not valid (preposition)
-When it is difficult to determine, the priority judgment is entity
Output format:
{
"component":"pronoun/preposition/conjunction/verb/adjective/adverb/incomplete sentence/general noun/abstract noun/named entity",
"valid": true/false
}
Example:
Text: the two hit by poisonous arrows were Dagger Devil Cao Zheng and Living Pluto 's Wife Wang Dingliu .
Phrase: Pluto
Answer:
```json
{"component": "named entity",
"valid": true}
```

B.2 Descriptor Generation Prompt Template

You are a helpful assistant.
Task* You are a professional text analysis assistant. Please process the input

text according to the following requirements: Task description:
1. for the given text and the specified phrase/word, generate 3-5 most relevant descriptions.
2. use commas to separate descriptors. Do not use periods or other signs.
3.The description can be given from the following aspects, the lowest level may include organization, location, person, art, event, product, other aspects.
4. the output must be in strict JSON format
`{"description": "description 1, description 2, description 3, ..."}`
Example:
Text: the two hit by poisonous arrows were Dagger Devil Cao Zheng and Living Pluto 's Wife Wang Dingliu .
Phrase: Pluto
Answer:
```json
{"description": "super natural, god, mythological figures"}```

### B.3 Make Final Decision From Candidates Prompt Template

You are a helpful assistant.
Task* Given a text, an entity, and a list of candidate labels, perform the following:
1. For each label in candidates, determine whether the entity belongs to that category.
2. Return a JSON list containing all the labels for which the entity was determined to belong. Like: ```json
`["label1", "label2", ...]` ```
Example:
Text: Thousands of residents of low - lying areas were ordered to evacuate as the storm headed north in the Gulf of Mexico with 80 mph winds .
Entity: the Gulf
Candidates:
`["/location", "/location/city", "/location/geography", "/location/geography/body_of_water"]`
Answer:
```json
["/location", "/location/geography", "/location/geography/body_of_water"]
```

C Candidate Selection Algorithm

Algorithm 1 Candidate Label Selection via Similarity Aggregation

Require: Entity descriptor set $D_e = \{d_e^{(1)}, d_e^{(2)}, \ldots, d_e^{(m)}\}$
Full label set $\mathcal{L}$
Label descriptors $\{D_l\}_{l \in \mathcal{L}}$
Parameters: q (top matches per descriptor), k (candidate size)
Ensure: Candidate label set $\mathcal{Y}_{\text{cand}}(e)$

Initialize score dictionary: $\text{Score}(l) \leftarrow 0$ for all $l \in \mathcal{L}$
for each entity descriptor $d_e^{(i)} \in D_e$ **do**
  Initialize similarity list $\mathcal{S} \leftarrow \emptyset$
  for each label $l \in \mathcal{L}$ **do**
    for each label descriptor $d_l \in D_l$ **do**
      Compute similarity: $s \leftarrow \cos(\mathbf{v}_{d_e^{(i)}}, \mathbf{v}_{d_l})$
      Add tuple (l, s) to $\mathcal{S}$
    end for
  end for
  Select top-q tuples from $\mathcal{S}$ by similarity: $\mathcal{P}_q^{(i)} \leftarrow \text{Top-}q(\mathcal{S})$
  for each tuple $(l^*, s) \in \mathcal{P}_q^{(i)}$ **do**
    $\text{Score}(l^*) \leftarrow \text{Score}(l^*) + s$
  end for
end for
Sort labels by score: $\mathcal{L}_{\text{sorted}} \leftarrow \text{sort}(\mathcal{L}, \text{Score}, \text{descending})$
Select top-k labels: $\mathcal{Y}_{\text{cand}}(e) \leftarrow$ first k labels in $\mathcal{L}_{\text{sorted}}$
return $\mathcal{Y}_{\text{cand}}(e)$

D Experimental Settings Details

Unless otherwise specified, all encoders used for descriptor retrieval are frozen during both training and inference. Entity descriptors are generated using the same base LLM as the classifier in a zero-shot or few-shot setting with fixed decoding parameters. For all fine-tuning experiments, FT-Raw and FT-Filtered share identical training configurations, and differ only in the construction of training data.

E Code Availability

The code used in this paper is available at an anonymous link:
```

Stage	Parameter	Value
Stage 2	Q	5
	K (OntoNotes)	10
	K (BBN)	10
Stage 3	Epochs	3
	Batch size	2
	Learning rate	2×10^{-4}
	LoRA r, α	8, 16
	Dropout	0.1
Environment	GPU	RTX 4090 (24GB)
	Base LLM	Qwen2.5-7B-Instruct
	Base Encoder	BERT
	Runtime (ms/sample)	210

Table 8: Key hyperparameters across training stages.

https://drive.google.com/file/d/1uYs_EXVS7tkXJ205c4LETFvC1XyGS9UW/view?usp=sharing

The provided code is modular and requires manual configuration for individual experiments. No single script reproduces all experiments automatically.