Type-enriched Hierarchical Contrastive Strategy for Fine-Grained Entity Typing

Xinyu Zuo, Haijin Liang, Ning Jing, Shuang Zeng, Zhou Fang and Yu Luo
Tencent Inc.
{xylonzuo,hodgeliang,shuangzeng,akirafang,yamiluo}@tencent.com
ning.jing.ustc@gmail.com

Abstract

Fine-grained entity typing (FET) aims to deduce specific semantic types of the entity mentions in text. Modern methods for FET mainly focus on learning what a certain type looks like. And few works directly model the type differences, that is, let models know the extent that one type is different from others. To alleviate this problem, we propose a type-enriched hierarchical contrastive strategy for FET. Our method can directly model the differences between hierarchical types and improve the ability to distinguish multi-grained similar types. On the one hand, we embed type into entity contexts to make type information directly perceptible. On the other hand, we design a constrained contrastive strategy on the hierarchical structure to directly model the type differences, which can simultaneously perceive the distinguishability between types at different granularity. Experimental results on three benchmarks, BBN, OntoNotes, and FIGER show that our method achieves significant performance on FET by effectively modeling type differences.

1 Introduction

Entity typing is a fundamental research problem in natural language processing (NLP), which aims to deduce the semantic types of the entity mentions in text. With the deepening of text understanding, the type sets of entities become more refined and ranging in from dozens (Hovy et al., 2006) to hundreds (Weischedel and Brunstein, 2005; Ling and Weld, 2012) or thousands (Choi et al., 2018). Therefore, fine-grained entity typing (FET) has gained more attention, which focuses on assigning more specific types to entities. For sentence in Figure 1, a FET system needs to assign a coarse-grained type *"/person"* and a fine-grained type *"/person/actor"* to the entity *"Vivien Leigh"*. The inferred fine-grained types could provide more specific prior knowledge for downstream NLP tasks, such as question answering (Lee et al., 2006) and entity linking (Leszczynski et al., 2022).

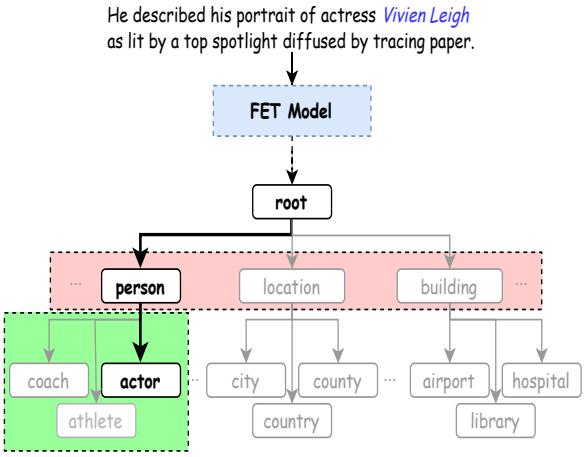

Figure 1: Example of fine-grained entity typing based on FIGER ontology. Green Box: the scope of visible types about *"person"* of the fine-grained contrastive strategy. Red Box: the scope of visible types of the coarse-grained contrastive strategy.

Considering the partial ontology of the FIGER dataset (Ling and Weld, 2012) in Figure 1, fine-grained entity types are often linked together in a hierarchical taxonomy, which makes the type boundaries increasingly blurred, especially for subtypes under the same coarse type. As shown in Figure 1, the fine-grained types *"coach"*, *"athlete"* and *"actor"*, all of which fall into the coarse-grained type *"person"*, are less differentiated.

In order to identify fine-grained types, prior work has concentrated on excavating more informative representations of types or entities, which benefiting from hand-crafted features (Ren et al., 2016), external resources (Onoe and Durrett, 2020; Li et al., 2022) or external pre-trained task (Xu et al., 2020). Most of them focus on learning *what a certain type looks like*, but few works have gone further to directly model the differences between types, that is, let models know *the extent that one*

Proceedings of the 29th International Conference on Computational Linguistics, pages 2405–2417
October 12–17, 2022.

type is different from others, which is more effective to distinguish among similar fine-grained types.

How to directly model the type differences? We argue that there are two key points, *Entity Type Awareness*, which refers to directly perceiving what type of entity is in the sentence, and *Type Differences Measure*, which refers to modeling how different the perceived type is from other types. For the first point, the intuitive idea is to expose the types directly, leading to a direct focus on what type the context represents. For the second point, heuristically, *direct is effective*, i.e., directly modeling which contexts represent the same types and which are different is the most efficient way to measure differences.

To this end, we propose a ty**P**e-enriched h**I**erarchical **CO**ntrastive stra**T**ey (**PICOT**) for fine-grained entity typing. Specifically, for *entity type awareness* with limited annotated data, inspired by prompt learning in entity typing (Ding et al., 2021), PICOT embeds the entity types in contexts via prompts to build type-rich expressions that guide the learning of correct types. Additionally, for *type differences measure*, PICOT takes a constrained contrastive strategy on hierarchical taxonomy to directly model the type differences from type-rich expressions. Concretely, as shown in Figure 1, PICOT is only concerned with the fine-grained types under the same coarse-grained type to learn the differences between fine-grained types. Similarly, PICOT is not concerned with what the fine-grained types are when distinguishing dissimilarities between coarse-grained types. Methodologically, PICOT learns the type differences at different granularity through type-rich expressions by limiting the scope of attention to types. Moreover, to further show models what a particular type is, we introduce a small number of type descriptions that directly expose richer type knowledge.

In experiments, we evaluate our model on three benchmarks. First, we concern with the standard evaluations and show that our model achieves the state-of-the-art performance on FET. Then we estimate the main components of PICOT. Finally, we do a visual analysis of the effectiveness of the type differentiation of PICOT.

In summary, the contributions are as follows:

- We propose a type-enriched hierarchical contrastive strategy (**PICOT**) for fine-grained entity typing. Our method can directly model the differences between hierarchical types and improve the ability to distinguish multi-grained similar types.
- First, we embed types into entity contexts to make type information directly perceptible. Then we design a constrained contrastive strategy on hierarchical taxonomy to directly model type differences at different granularities simultaneously.
- Experimental results on three benchmarks show that PICOT can achieve the SOTA performance on FET with limited annotated data.

2 Related Work

Entity Typing Named entity recognition (Tjong Kim Sang and De Meulder, 2003) and entity typing (Ling and Weld, 2012; Gillick et al., 2014) are fundamental research problems in NLP. Recently researchers pay more attention to fine-grained entity typing (FET) and ultra-fine entity typing (UFET) (Choi et al., 2018), which predict specific fine or ultra-fine types for given entities. To do so, obtaining more labeled data is the first research perspective, represented by distant supervision (Ling and Weld, 2012; Chen et al., 2019). With these, some researchers had focused on how to reduce noises in automatically labeled data (Gillick et al., 2014; Ren et al., 2016; Ren, 2020; Wu et al., 2019; Pan et al., 2022; Zhang et al., 2021b; Pang et al., 2022). Additionally, another key challenge is how to deal with hierarchical ontology. Most prior works regarded the hierarchical typing problem as a multi-label classification task and incorporated the hierarchical structure in different ways (Ren et al., 2016; Shimaoka et al., 2017; Xu and Barbosa, 2018; Murty et al., 2018; Chen et al., 2020b, 2022).

Some works attempted to mine more label information or better label representation. Abhishek et al. (2017) enhanced the label representation by sharing parameters; López and Strube (2020) embed types into a high-dimension; Xiong et al. (2019) introduced associated labels to enhance the label representation; Rabinovich and Klein (2017); Lin and Ji (2019) exploited co-occurrence structures and latent label representation; Additionally, several novel textual representations were applied to obtain richer entity contextual information, such as prompt based architecture (Ding et al., 2021) and box embeddings framework (Onoe et al., 2021).

Moreover, FET and UFET suffer from an obvious issue of the unseen types due to the lack of

annotated data. Therefore, a variety of paradigms were be studied to alleviate this issue (Huang et al., 2016; Ma et al., 2016; Obeidat et al., 2019; Zhou et al., 2018; Zhang et al., 2020; Chen et al., 2021). Moreover, some works further drew on different large-scale external data or knowledge to understand entity types (Onoe and Durrett, 2020; Xu et al., 2021; Dai et al., 2021; Li et al., 2022).

In summary, few prior works focus on directly modeling type differences. Therefore, this paper tries to let models know that one type is different from others without large-scale external resources. See Appendix B for more details.

Contrastive Learning Contrastive learning aims to further improve the model's ability to distinguish positive and negative examples, and has been a popular method for representation learning on computer vision tasks (Hjelm et al., 2018; Chen et al., 2020a; He et al., 2020). Recently, some researches have applied contrastive learning to natural language understanding tasks, aiming to obtain better text representations or to distinguish similar labels, such as the event causality identifier (Zuo et al., 2021), the contrastive self-supervised encoder (Fang and Xie, 2020), the supporting clustering framework (Zhang et al., 2021a), the abstractive summarization framework (Liu and Liu, 2021), the contrastive fine-tuning paradigm of pre-trained language for fine-grained text classification (Suresh and Ong, 2021), and so on. In this paper, we propose a constrained contrastive framework to directly model the hierarchical type differences.

Prompt Learning Prompt learning aims to leverage language prompts as contexts, and downstream tasks can be expressed as some cloze-style objectives similar to those pre-training objectives. Recently, a series of hand-crafted prompts have been widely used in natural language understanding (Liu et al., 2021b; Schick and Schütze, 2021; Feldman et al., 2019; Petroni et al., 2019; Trinh and Le, 2018; Ding et al., 2021). Moreover, to avoid expensive prompt design, automatic prompt has also been explored (Ren et al., 2016; Shin et al., 2020; Schick and Schütze, 2021), and some continuous prompts have also been proposed (Lester et al., 2021; Li and Liang, 2021). In this paper, we embed prompts to directly expose types in the entity contexts.

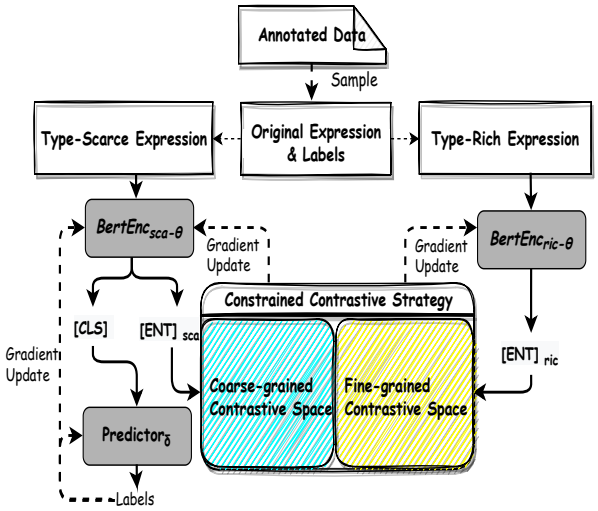

Figure 2: The framework of PICOT for FET (Sec. 4).

3 Problem Formulation

The input of fine-grained entity typing (FET) is a dataset $\mathcal{D} = \{x_1, x_2, ..., x_n\}$ with n sentences, a pre-defined hierarchical type ontology $\mathcal{Y}$, and each sentence x contains a marked entity e. A FET system is required to assign corresponding types to the given marked entity. Methodologically, for each input sequence $w_n = \{w_n^1, w_n^2, ..., w_n^t\}$ of the sentence x_n, FET aims to predict the correct multi-grained types $Y_n = \{y_n^1, y_n^2, ..., y_n^m\} \in \mathcal{Y}$ of the marked entity $e_n = \{w_n^l, ..., w_n^r\}$. For example in Figure 1, the correct type set of *"Vivien Leigh"* is $\{/person, /person/actor\}$, which contains a coarse-grained type "$/person$" and a fine-grained type "$/person/actor$".

4 Methodology

As shown in Figure 2, there are two key stages of PICOT for fine-grained entity typing.

- **Prompt-guided expression construction** (**ProExp**, Sec. 4.1). For *entity type awareness*, we construct two kinds of prompt-guided expressions, the *type-scarce expression* and the *type-rich expression*, which perceive the type patterns in context and expose type information directly, respectively.

- **Contrastive type knowledge transfer** (**ConTKT**, Sec. 4.2). For *type differences measure*, we propose a *constrained contrastive strategy* to directly model the differences among hierarchical types, and

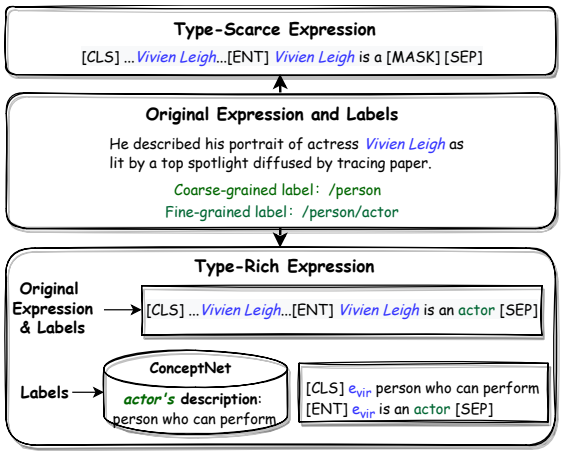

Figure 3: The illustration of prompt-guided expression construction (ProExp, Sec. 4.1).

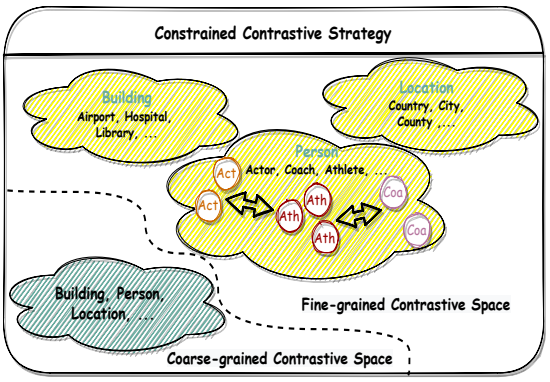

Figure 4: The illustration of constrained contrastive strategy in ConTKT (Sec. 4.2).

impart the type knowledge from type-rich expressions to predictor.

4.1 Prompt-guided Expression Construction (ProExp)

ProExp aims to convert the input sentences into type-scarce expressions and type-rich expressions based on entity-oriented prompts (Ding et al., 2021). The former could make models sensitive to type patterns in context, and the latter can be taken as the type knowledge resources based on type exposure.

Type-scarce Expression For each input sentence x_n, we construct type-scarce expression x_n^{ts} to guide the pre-trained language model (PLM, e.g. BERT (Devlin et al., 2019) used in this paper) encoder to efficiently exploit the entity contextual information, especially the type information. For simplicity, we choose declarative entity-oriented prompts to avoid grammatical errors.

Specifically, we first copy the marked entity e_n in x_n, then add a few conjunctions following the entity. Next, we add two specific words. One of them is *"[MASK]"* at the end of the expression, as a dummy for non-specific type. The other one is *"[ENT]"*, which bridges the original entity expression and prompt, and serves as an entry point for receiving type knowledge from type-rich expressions in ConTKT. The form of x_n^{ts} is as follows:

$$x_n^{ts} = x_n \textit{ [ENT] } e_n \textit{ is a [MASK].}$$

Type-rich Expression For each input sentence x_n, we also construct two kinds of type-rich expression x_n^{tr} as type knowledge resources for transfer in ConTKT when training. Intuitively, exposing the types directly to the context makes the expressions of entities type-aware.

Heuristically, fine-grained types contain both coarse- and fine- grained type properties. Therefore, we construct the type-rich expression x_n^{tr} of entity e_n by replacing the dummy type placeholder *"[MASK]"* in its type-scarce expression x_n^{ts} with its fine-grained types in Y_n[1]. Taking the entity in Figure 3 as an example, the form of x_n^{tr} is as follows:

$$x_n^{tr} = x_n \textit{ [ENT] } e_n \textit{ is an actor.}$$

Moreover, to better show what a particular type is, we introduce several descriptions (2 or 3) of each type from ConceptNet (Speer et al., 2017). Then we use them to directly expose richer type knowledge to construct extra type-rich expression x_{type}^{tr}. Specifically, for each fine-grained type, we replace x_n in x_n^{tr} with the combination of a virtual entity e_{vir} and one of descriptions. Taking the *actor* as an example:

$$x_{actor}^{tr} = e_{vir} \textit{ person who can perform [ENT] } e_{vir} \textit{ is an actor.}$$

4.2 Contrastive Type Knowledge Transfer (ConTKT)

ConTKT aims to directly model the differences among hierarchical types, and impart the type knowledge from type-rich expressions to predictor when training.

Expression Encoding We design two BERT encoders to encode two kinds of prompt-guided expressions for each entity e_n respectively. One is the encoder $BertEnc_{sca-\theta}$ for encoding type-scarce

[1]For entities with multiple fine-grained types, we concatenate the fine-grained types into one phrase by *"and"*.

expression x_n^{ts} when training and prediction, which digs out the type information of e_n in sentence x_n. Another is the encoder $BertEnc_{ric-\theta}$, which masters the type knowledge via encoding the type-rich expression x_n^{tr}.

Specifically, we first convert the x_n^{ts} and x_n^{tr} to the input sequences of two encoders respectively. Taking the example in Figure 3 as following:

$$\begin{aligned} w_n^{ts} =&[CLS], w_n^1, ..., w_n^t, [ENT], \\ & w_n^l, ..., w_n^r, is, a, [MASK], ., [SEP], \end{aligned} \tag{1}$$

$$\begin{aligned} w_n^{tr} =&[CLS], w_n^1, ..., w_n^t, [ENT], \\ & w_n^l, ..., w_n^r, is, an, actor, ., [SEP], \end{aligned} \tag{2}$$

$$\begin{aligned} w_{tr}^{actor} =&[CLS], des_{actor}^1, ..., des_{actor}^t, [ENT], \\ & w_{e_{vir}}, is, an, actor, ., [SEP]. \end{aligned} \tag{3}$$

where the des_{actor}^t is the token of type description and the $w_{e_{vir}}$ is the token of virtual entity e_{vir}.

After encoding, the representation $\boldsymbol{h}_n^{[CLS]_{ts}}$ of w_n^{ts} that encodes the contextual information of x_n^{ts} is used by predictor to predict types of entity e_n. Additionally, as mentioned above, the representation $\boldsymbol{h}_n^{[ENT]_{tr}}$ of w_n^{tr} is used as the exit of type knowledge contained in x_n^{tr}. Accordingly, the representation $\boldsymbol{h}_n^{[ENT]_{ts}}$ of w_n^{ts} is the entrance to receive the type knowledge from x_n^{tr} when training.

Constrained Contrastive Strategy We design a constrained contrastive strategy to directly model the hierarchical type differences based on prompt-guided expressions. Based on this, type knowledge is transferred to $BertEnc_{sca-\theta}$ from type-rich expressions by the contrast interaction between types at different granularities and the parameters sharing with $BertEnc_{ric-\theta}$.

Specifically, we only model the fine-grained type differences under the same coarse-grained type by bringing the same types closer while distancing different types. Likewise, for coarse-grained types, we do not care what the fine-grained types are in the same way. In this way, PICOT models the type differences between different granularities by limiting the scope of attention to types.

To specific, for one input batch $\mathcal{B} \subseteq \mathcal{D}$, there are two optimization objectives $\mathcal{L}_\theta^f$ and $\mathcal{L}_\theta^c$ to model the differences between fine-grained types and coarse-grained types respectively. And each optimization objective consists of two sub-optimization objectives, one for bringing the same types closer ($\mathcal{L}_\theta^{f+}$ and $\mathcal{L}_\theta^{c+}$), while another for distancing different types ($\mathcal{L}_\theta^{f-}$ and $\mathcal{L}_\theta^{c-}$):

$$\mathcal{L}_\theta^{f+} = \frac{1}{|\mathcal{Y}_\mathcal{B}^f|} \sum_{y \in \mathcal{Y}_\mathcal{B}^f} \frac{1}{2|\mathcal{B}^{f+}|} \sum_{x_i^y, x_j^y \in \mathcal{B}^{f+}}^{j \neq i} s(x_i^y, x_j^y), \tag{4}$$

$$\mathcal{L}_\theta^{f-} = -\frac{1}{2|\mathcal{B}^{f-}|} \sum_{x_i^{y_i}, x_j^{y_j} \in \mathcal{B}^{f-}}^{j \neq i, y_i \neq y_j} s(x_i^{y_i}, x_j^{y_j}), \tag{5}$$

$$\mathcal{L}_\theta^{c+} = \frac{1}{|\mathcal{Y}_\mathcal{B}^c|} \sum_{y \in \mathcal{Y}_\mathcal{B}^c} \frac{1}{2|\mathcal{B}^{c+}|} \sum_{x_i^y, x_j^y \in \mathcal{B}^{c+}}^{j \neq i} s(x_i^y, x_j^y), \tag{6}$$

$$\mathcal{L}_\theta^{c-} = -\frac{1}{2|\mathcal{B}^{c-}|} \sum_{x_i^{y_i}, x_j^{y_j} \in \mathcal{B}^{c-}}^{j \neq i, y_i \neq y_j} s(x_i^{y_i}, x_j^{y_j}), \tag{7}$$

$$s(x_i, x_j) = \lg \frac{e^{(dis(\boldsymbol{h}_i^E, \boldsymbol{h}_j^E)/\tau)}}{\sum_{x_i', x_j' \in \mathcal{B}*}^{j \neq i} e^{dis(\boldsymbol{h}_i^{E'}, \boldsymbol{h}_j^{E'})/\tau)}}, \tag{8}$$

where, take the $\mathcal{L}_\theta^f$ as illustration, $\mathcal{Y}_\mathcal{B}^f$ is the set of all fine-grained types in one batch, $y_\mathcal{B}^f$ is one of them. And $\mathcal{B}^{f+}$ consists of all x_n^{ts}, x_n^{tr} and x_{type}^{tr} with same fine-grained type $y \in \mathcal{Y}_\mathcal{B}^f$. Oppositely, $\mathcal{B}^{f-}$ consists of all x_n^{ts}, x_n^{tr} and x_{type}^{tr} with same coarse-grained type but different fine-grained types $y_i, y_j \in \mathcal{Y}_\mathcal{B}^f$. Moreover, s is the similarity between x_i and x_j, $\boldsymbol{h}_i^E$ and $\boldsymbol{h}_j^E$ are the $\boldsymbol{h}_n^{[ENT]}$ of x_i and x_j after encoding respectively, dis is the ℓ_2-distance function to measure the distance of two representation, τ is a temperature that adjusts the concentration level and $\mathcal{B}*$ is $\mathcal{B}^{f+}$ or $\mathcal{B}^{f-}$. Likewise, the optimization objectives are similar for $\mathcal{L}_\theta^c$.

Learning of FET After transferring, the $\boldsymbol{h}_n^{[CLS]_{ts}}$ of x_n^{ts} output by $BertEnc_{sca-\theta}$, which has learned types knowledge, is fed to the predictor to identify the types of the input e_n as following:

$$y_n^* \leftarrow MLP(\boldsymbol{h}_n^{[CLS]_{ts}}), \tag{9}$$

where y_n^* is the predicted types of e_n in x_n.

For training of two encoders and predictor, we add the constrained contrastive losses $\mathcal{L}_\theta^f$ and $\mathcal{L}_\theta^c$ to the classification loss $\mathcal{L}_\delta$. Finally, we minimize

Algorithm 1 Learning of PICOT for FET.

Require: type-scarce expression x^{ts}, type-rich expression x^{tr}, extra type-rich expression x^{tr}_{type}.
Training:
Stage: PROMPT-GUIDED EXPRESSION CONSTRUCTION
  for each batch $\mathcal{B} \in \mathcal{D}$ **do**
    for input entity e_n with its sentence $x_n \in \mathcal{B}$ **do**
      Construct type-scarce expression x^{ts}_n;
      Construct type-rich expression x^{tr}_n;
    end for

    for each fine-grained $type \in \mathcal{Y}_\mathcal{B}$ **do**
      Construct extra type-rich expression x^{tr}_{type};
    end for
  end for
end Stage:

Stage: CONTRASTIVE TYPE KNOWLEDGE TRANSFER
  for each batch $\mathcal{B} \in \mathcal{D}$ **do**
    for each fine-grained type y_f in $\mathcal{Y}_\mathcal{B}$ **do**
      Compute $\mathcal{L}^f_\theta$ in equation (4) and (5);
    end for

    for each coarse-grained type y_c in $\mathcal{Y}_\mathcal{B}$ **do**
      Compute $\mathcal{L}^c_\theta$ in equation (6) and (7);
    end for

    Compute batch classification loss $\mathcal{L}_\delta$ in (10);
    Compute $\mathcal{L}$ in equation (11);
    Stochastic gradient update θ and δ in (12);
  end for
end Stage:

the L and stochastic gradient update the θ and δ as Algorithm 1:

$$\mathcal{L}_\delta = BCEWithLogits(y^*_n, y_n), \tag{10}$$

$$\begin{aligned}\mathcal{L} = \mathcal{L}_\delta &+ \lambda_f(\mathcal{L}^{f+}_\theta + \mathcal{L}^{f-}_\theta) \\ &+ \lambda_c(\mathcal{L}^{c+}_\theta + \mathcal{L}^{c-}_\theta),\end{aligned} \tag{11}$$

$$\theta, \delta \leftarrow \eta \nabla \mathcal{L}, \tag{12}$$

where, λ_f and λ_c are the weights of $\mathcal{L}^f_\theta$ and $\mathcal{L}^c_\theta$ respectively, η is the learning rate.

5 Experiments

5.1 Experimental Setup

Datasets and Evaluation Metrics We conduct experiments on three standard FET datasets and follow the version processed and split by Onoe et al. (2021). (1) **BBN** (Weischedel and Brunstein, 2005), which contains 56 types and each type has a maximum type hierarchy level of 2; (2) **OntoNotes** (Gillick et al., 2014), which is sampled from the OntoNotes (Weischedel et al., 2013) corpus and re-annotated with 89 types in 3-level hierarchy. Additionally, we ignore the *other* type, which has no obvious meaning, and categorize it into two-level types; (3) **FIGER** (Ling and Weld, 2012), which contains 113 types and each type also has a maximum type hierarchy level of 2. Table 1 is the statistics on the coarse-grained and fine-grained types of three datasets. Moreover, we evaluate three datasets using the standard metrics: Macro F1 (Ma-F1) and Micro F1 (Mi-F1).

Datasets	#Coarse	#Fine	#Fine/Coarse
BBN	17	39	2.3
OntoNotes	20	68	3.4
FIGER	47	66	1.4

Table 1: Statistics on the coarse-grained and fine-grained types of three datasets.

Parameters Settings For a fair comparison, similar to Onoe et al. (2021), the BERT encoders are BERT-Large architecture[2], which has 24-layers, 1024-hiddens, and 16-heads. For parameters, we set the learning rate of η as 8e-6, and set the temperature τ of the contrastive loss as 0.1 tuned on the development set. Moreover, we also tune the batch size to 96 on the development set. The λ_f and λ_c are setted as 0.1/0.1/0.1 and 0.1/0.1/0.01/ for three datasets respectively. And we apply the early stop and AdamW gradient strategy to optimize all models. Additionally, to simulate constraints like the previous work (Chen et al., 2020b; Onoe et al., 2021), we use the same three simple rules to modify the model's predictions or training data on BBN datasets: (1) dropping *"person"* if *"organization"* exists, (2) dropping *"location"* if *"gpe"* exists, and (3) replacing *"facility"* by *"fac"*, since both the two tags appear in the training set but only *"fac"* in the test set. See Appendix C for more detailed settings.

Compared Methods Same as previous methods, we prefer the following models which use the same versions of three datasets and do not rely on large-scale external knowledge or resources as our compared methods[3]. (1) **Ren et al. (2016)**, a embedding method which separately models clean and noisy data with type hierarchy; (2) **Abhishek et al. (2017)**, a neural network model that jointly learns entities and their contexts representation; (3) **Zhang et al. (2018)**, a neural architecture which leverages both document and sentence level infor-

[2] https://github.com/google-research/bert

[3] There are different versions of three datasets exist.

Methods	BBN		OntoNotes		FIGER	
	Ma-F1	Mi-F1	Ma-F1	Mi-F1	Ma-F1	Mi-F1
Ren et al. (2016)	74.1	75.7	71.1	64.7	69.3	66.4
Abhishek et al. (2017)	74.1	75.7	68.5	63.3	78.0	74.9
Zhang et al. (2018)	75.7	75.1	72.1	66.5	78.7	75.5
Chen et al. (2020b) (exclusive)	63.2	61.0	72.4	67.2	82.6	**80.8**
Chen et al. (2020b) (undefined)	79.7	80.5	73.0	68.1	80.5	78.1
Lin and Ji (2019)	79.3	78.1	82.9*	77.3*	83.0	79.8
Onoe et al. (2021) (vector)	78.3	78.0	76.2	68.9	81.6	77.0
Onoe et al. (2021) (box)	78.7	78.0	77.3	70.9	79.4	75.0
Liu et al. (2021a)	-	-	77.6	71.8	-	-
PICOT	**81.8**	**82.2**	**78.7**	**72.1**	**84.7**	79.6

Table 2: Results on fine-grained entity typing. *: Not directly comparable since large-scale augmentated data is used. The results are tested for significance at the 0.05 level.

Methods	BBN		
	Ma-F1	Mi-F1	∇
PICOT (our)	**81.8**	**82.8**	-
w/o Exp.$_{tr_{des}}$	81.6	82.2	-0.2/-0.6
w/o Exp.$_{tr}$	81.4	81.7	-0.4/-1.1
w/o Exp.$_{ts\&tr}$	81.1	81.5	-0.7/-1.3
Previous SOTA	79.7	80.5	-

Table 3: Ablation results of the prompt-guided expression (ProExp, Sec. 4.1) of FET on BBN. w/o Exp.$_{tr_{des}}$ denotes a varietal PICOT that without extra type-rich expression when training; w/o Exp.$_{tr}$ denotes a varietal PICOT that without all type-rich expressions when training; w/o Exp.$_{ts\&tr}$ denotes a varietal PICOT that without all prompt-guided expressions when training.

Methods	BBN		
	Ma-F1	Mi-F1	∇
PICOT (our)	**81.8**	**82.8**	-
w/o ConTKT$_{coar}$	80.0	80.4	-1.8/-2.4
w/o ConTKT$_{fine}$	80.5	81.0	-1.3/-1.8
w/o ConTKT	80.2	80.7	-1.6/-2.1

Table 4: Ablation results of the contrastive type knowledge transfer (ConTKT, Sec. 4.2) of FET on BBN. w/o ConTKT$_{coar}$ denotes a varietal PICOT that removes coarse-grained contrastive loss; w/o ConTKT$_{fine}$ denotes a varietal PICOT that removes fine-grained contrastive loss; w/o ConTKT denotes a varietal PICOT that removes the whole ConTKT.

mation; (4) **Chen et al. (2020b) (exclusive)**, a classifier for hierarchical FET that embraces ontological structure with exclusive interpretations; (5) **Chen et al. (2020b) (undefined)**, a same classifier as (4) but with different undefined interpretations; (6) **Lin and Ji (2019)**, a FET model with a novel attention mechanism and a hybrid type classifier; (7) **Onoe et al. (2021) (vector)**, a vector-based model for FET; (8) **Onoe et al. (2021) (box)**, a box-based model for FET; (9) **Liu et al. (2021a)**, a FET model with extrinsic and intrinsic dependencies between labels. Moreover, all results of compared methods are directly copied from the previous papers.

5.2 Our Method vs. State-of-the-art Methods

Table 2 shows the results of FET on BBN, OntoNotes, and FIGER. From the results, we can observe that (see Appendix A for more results):

(1) On BBN, our PICOT outperforms all baselines and achieves the best performance on Macro F1 and Micro F1 values, which are 81.8% and 82.2%, outperforming the state-of-the-art by a margin of 2.1% and 1.7% respectively, which justifies its effectiveness. Moreover, among the three datasets, the BBN dataset has the least training data but the largest boost. This indicates that PICOT can effectively mine type information in limited labeled data by sensing type knowledge and directly modeling the differences between types.

(2) On OntoNotes, compared with the methods without large external data, our PICOT also achieves the best performance on Macro F1 and Micro F1 values, which are 78.7% and 72.1%, outperforming by a margin of 1.1% and 0.3% respectively. Although OntoNotes has three times more training data than BBN and can provide more type information to compared models, the proposed PICOT can still further improve the performance, which demonstrates the effectiveness of directly modeling type differences.

(3) On FIGER, the largest one in three datasets,

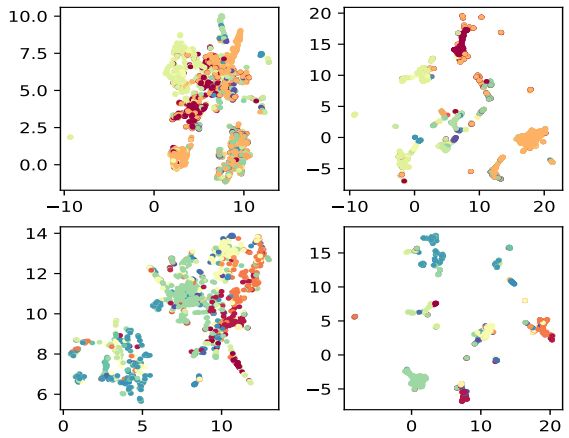

Figure 5: The visualization of type representation clustering without- (left) and with- (right) PICOT on development dataset. Specifically, top row is coarse-grained type clustering and bottom row is fine-grained type clustering. Each color represents a kind of type

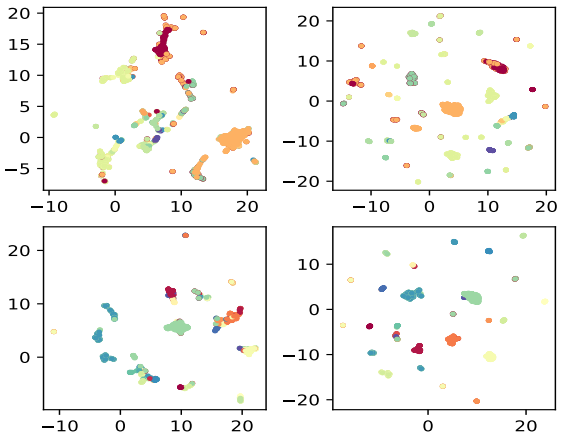

Figure 6: The visualization of type representation clustering of type-scarce (left) and type-rich (right) expressions on development dataset. Specifically, top row is coarse-grained type and bottom row is fine-grained type. Each color represents a kind of type

our PICOT outperforms the best compared method on Macro-F1 value by a margin of 1.7%, which further proves the effectiveness of PICOT in mining type differences with labeled data. It is worth noting that FIGER has a slightly lower performance on the Micro-F1 value due to the inconsistent of some test samples, in which only have fine-grained types (e.g., *"/organization/sports_team"* is present, but *"/organization"* is missing).

5.3 Effect of Prompt-guided Expression

We analyze the effect of the prompt-guided expression (ProExp, Sec. 4.1) on BBN dataset. As shown in Table 3, from the results, we can observe that: (1) after removing the type-rich expressions (***w/o Exp.***$_{tr_{des}}$ and ***w/o Exp.***$_{tr}$), the performance of FET significantly decreases. This proves that exposing the type information directly to the model can bring great help to determine entity types. (2) Comparing the ***w/o Exp.***$_{tr_{des}}$ with the ***previous SOTA***, we find that without introducing any external type descriptions, PICOT could also effectively mine type knowledge within the limited labeled data and improve the performance of FET. (3) Comparing the ***w/o Exp.***$_{tr_{des}}$ with ***w/o Exp.***$_{tr}$, just with a small amount of external types descriptions, our PICOT's type knowledge exposure and transfer framework also enhance the performance. (4) ***w/o Exp.***$_{ts\&tr}$ also achieves good results without any prompt-guided expressions, which also shows the effectiveness of the contrastive transfer strategy.

5.4 Effect of Contrastive Type Knowledge Transfer

We analyze the effect of the contrastive type knowledge transfer (ConTKT, Sec. 4.2) on the BBN dataset. As shown in Table 4, from the results, we can observe that: (1) after removing the ConTKT (***w/o ConTKT***), the performance of FET significantly decreases. This illustrates that the contrastive strategy can effectively improve the discrimination of similar types, which is important for FET. (2) Comparing ***w/o ConTKT***$_{coar}$, ***w/o ConTKT***$_{fine}$ and ***PICOT***, we find that both coarse-grained and fine-grained contrastive training play a key role in the measurement of type differences. (3) It is worth noting that, comparing ***w/o ConTKT***$_{coar}$ with ***w/o ConTKT***, we find that training with type-scarce expression without contrastive strategy works better than only using fine-grained type contrastive strategy. Meanwhile, coarse-grained contrastive training alone (w/o ***ConTKT***$_{fine}$) only give a small boost for FET. These indicate that only the combination of coarse-grained and fine-grained contrastive strategies can achieve the desired results. Specifically, coarse-grained contrast ensures the base performance while fine-grained contrast further improves the ability to discriminate types.

5.5 Visualization of the Effect of Type Distinguishing

To further illustrate the effect of PICOT, in Figure 5, we cluster the representations of *"[ENT]"* in the type-scarce expressions before and after train-

ing by UMAP downscaling (McInnes et al., 2018). The comparisons of the left and right subgraphs show that the differentiation of *"[ENT]"* representations, which is the entrance for type knowledge, is both greatly improved by PICOT among the coarse-grained and fine-grained types. This illustrates that PICOT can effectively improve the model's ability to discriminate against similar types.

As shown in Figure 6, to elucidate the effect of direct type exposure for type differentiation, we cluster the *"[CLS]"* representations of type-scarce and type-rich expressions, respectively. The comparisons show that the representation of type-rich expressions is more discriminative, especially for fine-grained types, which can effectively guide the model to identify types with high similarity.

6 Conclusion

We propose a type-enriched hierarchical contrastive strategy for fine-grained entity typing. Our method can directly model the differences between hierarchical types and improve the ability to distinguish multi-grained similar types. First, we embed types into entity contexts to make type information directly perceptible. Then we design a constrained contrastive strategy on hierarchical taxonomy to directly model type differences at different granularities simultaneously. Experimental results on three benchmarks show that PICOT can achieve state-of-the-art performance on FET with limited annotated data.

References

Abhishek Abhishek, Ashish Anand, and Amit Awekar. 2017. Fine-grained entity type classification by jointly learning representations and label embeddings. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 797–807, Valencia, Spain. Association for Computational Linguistics.

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. *arXiv preprint arXiv:1904.06475*.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020a. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

Tongfei Chen, Yunmo Chen, and Benjamin Van Durme. 2020b. Hierarchical entity typing via multi-level learning to rank. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8465–8475, Online. Association for Computational Linguistics.

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 *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2076–2087, Dublin, Ireland. Association for Computational Linguistics.

Yi Chen, Haiyun Jiang, Lemao Liu, Shuming Shi, Chuang Fan, Min Yang, and Ruifeng Xu. 2021. An empirical study on multiple information sources for zero-shot fine-grained entity typing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2668–2678, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Eunsol Choi, Omer Levy, Yejin Choi, and Luke Zettlemoyer. 2018. Ultra-fine entity typing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 87–96, Melbourne, Australia. Association for Computational Linguistics.

Hongliang Dai, Yangqiu Song, and Haixun Wang. 2021. Ultra-fine entity typing with weak supervision from a masked language model. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1790–1799, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ning Ding, Yulin Chen, Xu Han, Guangwei Xu, Pengjun Xie, Hai-Tao Zheng, Zhiyuan Liu, Juanzi Li, and Hong-Gee Kim. 2021. Prompt-learning for fine-grained entity typing. *arXiv preprint arXiv:2108.10604*.

Hongchao Fang and Pengtao Xie. 2020. CERT: contrastive self-supervised learning for language understanding. *CoRR*, abs/2005.12766.

Joshua Feldman, Joe Davison, and Alexander M. Rush. 2019. Commonsense knowledge mining from pretrained models. *CoRR*, abs/1909.00505.

Dan Gillick, Nevena Lazic, Kuzman Ganchev, Jesse Kirchner, and David Huynh. 2014. Context-dependent fine-grained entity type tagging. *arXiv preprint arXiv:1412.1820*.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning.

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. 2018. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*.

Eduard Hovy, Mitch Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. Ontonotes: the 90% solution. In *Proceedings of the human language technology conference of the NAACL, Companion Volume: Short Papers*, pages 57–60.

Lifu Huang, Jonathan May, Xiaoman Pan, and Heng Ji. 2016. Building a fine-grained entity typing system overnight for a new x (x= language, domain, genre). *arXiv preprint arXiv:1603.03112*.

Changki Lee, Yi-Gyu Hwang, Hyo-Jung Oh, Soojong Lim, Jeong Heo, Chung-Hee Lee, Hyeon-Jin Kim, Ji-Hyun Wang, and Myung-Gil Jang. 2006. Fine-grained named entity recognition using conditional random fields for question answering. In *Asia information retrieval symposium*, pages 581–587. Springer.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *CoRR*, abs/2104.08691.

Megan Leszczynski, Daniel Y Fu, Mayee F Chen, and Christopher Ré. 2022. Tabi: Type-aware bi-encoders for open-domain entity retrieval. *arXiv preprint arXiv:2204.08173*.

Bangzheng Li, Wenpeng Yin, and Muhao Chen. 2022. Ultra-fine entity typing with indirect supervision from natural language inference. *arXiv preprint arXiv:2202.06167*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *CoRR*, abs/2101.00190.

Ying Lin and Heng Ji. 2019. An attentive fine-grained entity typing model with latent type representation. In *EMNLP-IJCNLP 2019*, pages 6197–6202, Hong Kong, China. Association for Computational Linguistics.

Xiao Ling and Daniel S Weld. 2012. Fine-grained entity recognition. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*.

Qing Liu, Hongyu Lin, Xinyan Xiao, Xianpei Han, Le Sun, and Hua Wu. 2021a. Fine-grained entity typing via label reasoning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4611–4622, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021b. GPT understands, too. *CoRR*, abs/2103.10385.

Yixin Liu and Pengfei Liu. 2021. SimCLS: A simple framework for contrastive learning of abstractive summarization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 1065–1072, Online. Association for Computational Linguistics.

Federico López and Michael Strube. 2020. A fully hyperbolic neural model for hierarchical multi-class classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 460–475, Online. Association for Computational Linguistics.

Yukun Ma, Erik Cambria, and Sa Gao. 2016. Label embedding for zero-shot fine-grained named entity typing. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 171–180, Osaka, Japan. The COLING 2016 Organizing Committee.

Leland McInnes, John Healy, and James Melville. 2018. Umap: Uniform manifold approximation and projection for dimension reduction.

Shikhar Murty, Patrick Verga, Luke Vilnis, Irena Radovanovic, and Andrew McCallum. 2018. Hierarchical losses and new resources for fine-grained entity typing and linking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 97–109, Melbourne, Australia. Association for Computational Linguistics.

Rasha Obeidat, Xiaoli Fern, Hamed Shahbazi, and Prasad Tadepalli. 2019. Description-based zero-shot fine-grained entity typing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 807–814, Minneapolis, Minnesota. Association for Computational Linguistics.

Yasumasa Onoe, Michael Boratko, Andrew McCallum, and Greg Durrett. 2021. Modeling fine-grained entity types with box embeddings. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2051–2064, Online. Association for Computational Linguistics.

Yasumasa Onoe and Greg Durrett. 2020. Fine-grained entity typing for domain independent entity linking.

In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8576–8583.

Weiran Pan, Wei Wei, and Feida Zhu. 2022. Automatic noisy label correction for fine-grained entity typing. *arXiv preprint arXiv:2205.03011*.

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.

Fabio Petroni, Tim Rocktäschel, Patrick S. H. Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? *CoRR*, abs/1909.01066.

Maxim Rabinovich and Dan Klein. 2017. Fine-grained entity typing with high-multiplicity assignments. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 330–334, Vancouver, Canada. Association for Computational Linguistics.

Quan Ren. 2020. Fine-grained entity typing with hierarchical inference. In *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, volume 1, pages 2552–2558. IEEE.

Xiang Ren, Wenqi He, Meng Qu, Lifu Huang, Heng Ji, and Jiawei Han. 2016. AFET: Automatic fine-grained entity typing by hierarchical partial-label embedding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1369–1378, Austin, Texas. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.

Sonse Shimaoka, Pontus Stenetorp, Kentaro Inui, and Sebastian Riedel. 2017. Neural architectures for fine-grained entity type classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1271–1280, Valencia, Spain. Association for Computational Linguistics.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *CoRR*, abs/2010.15980.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-first AAAI conference on artificial intelligence*.

Varsha Suresh and Desmond Ong. 2021. Not all negatives are equal: Label-aware contrastive loss for fine-grained text classification. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4381–4394, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Trieu H. Trinh and Quoc V. Le. 2018. A simple method for commonsense reasoning. *CoRR*, abs/1806.02847.

Ralph Weischedel and Ada Brunstein. 2005. Bbn pronoun coreference and entity type corpus. *Linguistic Data Consortium, Philadelphia*, 112.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. 2013. Ontonotes release 5.0 ldc2013t19. *Linguistic Data Consortium, Philadelphia, PA*, 23.

Junshuang Wu, Richong Zhang, Yongyi Mao, Hongyu Guo, and Jinpeng Huai. 2019. Modeling noisy hierarchical types in fine-grained entity typing: A content-based weighting approach. In *IJCAI*, pages 5264–5270.

Wenhan Xiong, Jiawei Wu, Deren Lei, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. 2019. Imposing label-relational inductive bias for extremely fine-grained entity typing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 773–784, Minneapolis, Minnesota. Association for Computational Linguistics.

Peng Xu and Denilson Barbosa. 2018. Neural fine-grained entity type classification with hierarchy-aware loss. *arXiv preprint arXiv:1803.03378*.

Zenan Xu, Daya Guo, Duyu Tang, Qinliang Su, Linjun Shou, Ming Gong, Wanjun Zhong, Xiaojun Quan, Nan Duan, and Daxin Jiang. 2020. Syntax-enhanced pre-trained model. *arXiv preprint arXiv:2012.14116*.

Zenan Xu, Daya Guo, Duyu Tang, Qinliang Su, Linjun Shou, Ming Gong, Wanjun Zhong, Xiaojun Quan, Daxin Jiang, and Nan Duan. 2021. Syntax-enhanced

pre-trained model. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5412–5422, Online. Association for Computational Linguistics.

Dejiao Zhang, Feng Nan, Xiaokai Wei, Shang-Wen Li, Henghui Zhu, Kathleen R. McKeown, Ramesh Nallapati, Andrew O. Arnold, and Bing Xiang. 2021a. Supporting clustering with contrastive learning. *CoRR*, abs/2103.12953.

Haoyu Zhang, Dingkun Long, Guangwei Xu, Muhua Zhu, Pengjun Xie, Fei Huang, and Ji Wang. 2021b. Learning with noise: improving distantly-supervised fine-grained entity typing via automatic relabeling. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 3808–3815.

Sheng Zhang, Kevin Duh, and Benjamin Van Durme. 2018. Fine-grained entity typing through increased discourse context and adaptive classification thresholds. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 173–179, New Orleans, Louisiana. Association for Computational Linguistics.

Tao Zhang, Congying Xia, Chun-Ta Lu, and Philip Yu. 2020. MZET: Memory augmented zero-shot fine-grained named entity typing. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 77–87, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Ben Zhou, Daniel Khashabi, Chen-Tse Tsai, and Dan Roth. 2018. Zero-shot open entity typing as type-compatible grounding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2065–2076, Brussels, Belgium. Association for Computational Linguistics.

Xinyu Zuo, Pengfei Cao, Yubo Chen, Kang Liu, Jun Zhao, Weihua Peng, and Yuguang Chen. 2021. Improving event causality identification via self-supervised representation learning on external causal statement. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2162–2172, Online. Association for Computational Linguistics.

A Supplementary Experiment Results

A.1 Effect of the Weights of Different Contrastive Loss

Coarse / Fine	**0.01**	**0.1**	**0.5**
0.01	-	81.6	-
0.1	81.4	**81.8**	81.3
0.5	-	78.5	-

Table 5: Macro F1 of PICOT on BBN with different coarse- (λ_c) and fine-grained (λ_f) contrastive weights.

Coarse / Fine	**0.01**	**0.1**	**0.5**
0.01	-	82.1	-
0.1	82.0	**82.8**	81.8
0.5	-	79.9	-

Table 6: Micro F1 of PICOT on BBN with different coarse- (λ_c) and fine-grained (λ_f) contrastive weights.

To further explore the effect of contrastive loss of different granularities, we vary the weights of fine- and coarse-grained contrastive loss to observe the performance of PICOT on the BBN test set, respectively. As shown in Table 5 and 6, we can notice that the type knowledge is not fully migrated when the coarse-grained and fine-grained contrastive loss weights are too small, and overly affects the classification performance when the weights are too large. It is worth noting that excessive fine-grained contrastive loss weights significantly degrade the performance because many fine-grained types are not completely distinct, and some types could occur simultaneously. Therefore, excessive differentiation of fine-grained types will confuse models.

A.2 Effect of the *"[ENT]"* Position

Pos.	Ma-F1	Mi-F1
After *"[CLS]"*	80.7	81.2
Before prompt	**81.8**	**82.8**

Table 7: Performance of PICOT with different *"[ENT]"* positions.

As shown in Table 7, we further explore the effect of the position of *"[ENT]"* as type knowledge exit and entry in the input sequence on the PICOT

performance. From the results, we can see that placing *"[ENT]"* between the entity context and the type prompt allows for more efficient migration and reception of type knowledge.

B Supplementary Related Work

Named entity recognition (Tjong Kim Sang and De Meulder, 2003) and entity typing (Ling and Weld, 2012; Gillick et al., 2014) are fundamental research problems in NLP. Recently researchers pay more attention on fine-grained entity typing (FET) and ultra-fine entity typing (UFET) (Choi et al., 2018), which predicts specific fine or ultra-fine types for given entities. To do so, obtaining more labeled data is the first research perspective for FET, represented by the distant supervision annotation method (Ling and Weld, 2012). With these, some researches had focused on how to reduce noises in automatically labeled data, such as a heuristic constraint pruning approach (Gillick et al., 2014), a partial-label loss (Ren et al., 2016), a penalty optimization term (Ren, 2020), and a novel content-sensitive weighting schema (Wu et al., 2019).

Additionally, one key challenge is how to deal with hierarchical type ontology. Most prior works regarded the hierarchical typing problem as a multi-label classification task and incorporated the hierarchical structure in different ways. Ren et al. (2016) used a predefined label hierarchy to reduce noises; Shimaoka et al. (2017) proposed a hierarchical label encoding method; Xu and Barbosa (2018) employed a normalized hierarchical loss; Murty et al. (2018) learned a subtyping relation to constrain the type embedding; Chen et al. (2020b) designed a novel loss function to exploit label hierarchies.

Some work attempted to mine more label information or better label representation. Abhishek et al. (2017) enhanced the label representation by sharing parameters; López and Strube (2020) embed types into a high-dimension; Xiong et al. (2019) introduced associated labels to enhance the label representation; Rabinovich and Klein (2017) exploited co-occurrence structures during label set prediction; (Lin and Ji, 2019) reconstructed the co-occurrence structure via latent label representation; Liu et al. (2021a) reasoned fine-grained types by discovering label dependencies knowledge. Additionally, several novel textual representations were applied to obtain richer entity contextual information. Ding et al. (2021) investigated the application of prompt-learning to predict fine-grained entity types. Onoe et al. (2021) studied the box embeddings to capture hierarchies of types.

Moreover, FET and UFET suffer from an obvious issue of the unseen types due to the lack of annotated data. Therefore, a variety of paradigms were being studied to alleviate this issue, such as a hierarchical clustering model (Huang et al., 2016), a prototypical embedding method (Ma et al., 2016), a context-description matching model based on type descriptions from Wikipedia (Obeidat et al., 2019), a classifier based on Freebase types of its type-compatible, (Zhou et al., 2018), a novel framework which transfers the knowledge from seen types to the unseen ones (Zhang et al., 2020), and an empirical study on multiple auxiliary information (Chen et al., 2021). To further alleviate the lack of annotated data, some work draws on different large-scale external data or knowledge to understand the types of entities in the sentences. Onoe and Durrett (2020) used hyperlinked mentions in Wikipedia to distantly label large scale data and train an entity typing model; Xu et al. (2021) introduced a new pre-training task of predicting the syntactic distance in dependency tree based on large scale texts; Dai et al. (2021) automatically generated new ultra-fine entity typing data with labels; Li et al. (2022) presented LITE, a new approach that formulates entity typing as an NLI problem based on external data.

In summary, few prior works focus on directly modeling the differences between types. Therefore, this paper tries to let models know that one type is different from others without large-scale external resources.

C Main Experimental Environments and Other Parameters Settings

C.1 Experimental Environments

We deploy all models on a server with Tesla P40 GPU. Specifically, the configuration environment of the server is ubuntu 16.04, and our framework mainly depends on python 3.8.8 and Torch 1.11.

C.2 Other Parameters Settings

All the final hyper-parameters for evaluation are averaged after 3 independent tunings on the development set. Moreover, the three datasets BBN, OntoNotes, and FIGER achieve optimal results at the 20th/10th/5th epochs, which take half a day, one day, and two days, respectively.

This is an appendix.