Fine-grained Entity Typing via Label Reasoning

Qing Liu[1,3], **Hongyu Lin**[1]*, **Xinyan Xiao**[4], **Xianpei Han**[1,2]*, **Le Sun**[1,2], **Hua Wu**[4]
[1]Chinese Information Processing Laboratory [2]State Key Laboratory of Computer Science
Institute of Software, Chinese Academy of Sciences, Beijing, China
[3]University of Chinese Academy of Sciences, Beijing, China
[4]Baidu Inc., Beijing, China
{liuqing2020,hongyu,xianpei,sunle}@iscas.ac.cn
{xiaoxinyan,wu_hua}@baidu.com

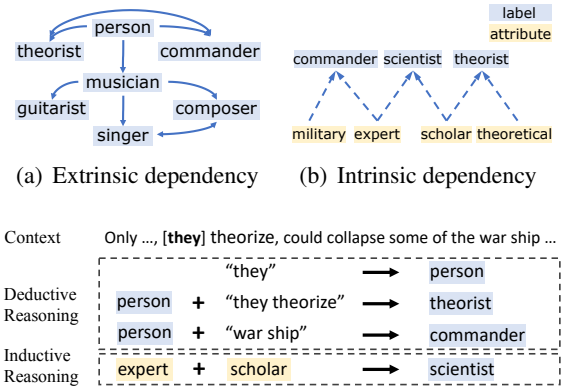
Abstract

Conventional entity typing approaches are based on independent classification paradigms, which make them difficult to recognize inter-dependent, long-tailed and fine-grained entity types. In this paper, we argue that the implicitly entailed extrinsic and intrinsic dependencies between labels can provide critical knowledge to tackle the above challenges. To this end, we propose *Label Reasoning Network(LRN)*, which sequentially reasons fine-grained entity labels by discovering and exploiting label dependencies knowledge entailed in the data. Specifically, LRN utilizes an auto-regressive network to conduct deductive reasoning and a bipartite attribute graph to conduct inductive reasoning between labels, which can effectively model, learn and reason complex label dependencies in a sequence-to-set, end-to-end manner. Experiments show that LRN achieves the state-of-the-art performance on standard ultra fine-grained entity typing benchmarks, and can also resolve the long tail label problem effectively.

1 Introduction

Fine-grained entity typing (FET) aims to classify entity mentions to a fine-grained semantic label set, e.g., classify "*FBI agents*" in "*They were arrested by* ***FBI agents***." as {*organization, administration, force, agent, police*}. By providing fine-grained semantic labels, FET is critical for entity recognition (Lin et al., 2019a,b, 2020; Zhang et al., 2021b,a) and can benefit many NLP tasks, such as relation extraction (Yaghoobzadeh et al., 2017; Zhang et al., 2019), entity linking (Onoe and Durrett, 2020) and question answering (Yavuz et al., 2016).

The fundamental challenge of FET comes from its large-scale and fine-grained entity label set, which leads to significant difference between FET and conventional entity typing. First, due to the massive label set, it is impossible to independently recognize each entity label without considering their dependencies. For this, existing approaches use the predefined label hierarchies (Ren et al., 2016a; Shimaoka et al., 2017; Abhishek et al., 2017; Karn et al., 2017; Xu and Barbosa, 2018; Wu et al., 2019; Chen et al., 2020; Ren, 2020) or label co-occurrence statistics from training data (Rabinovich and Klein, 2017; Xiong et al., 2019; Lin and Ji, 2019) as external constraints. Unfortunately, these label structures or statistics are difficult to obtain when transferring to new scenarios. Second, because of the fine-grained and large-scale label set, many long tail labels are only provided with several or even no training instances. For example, in Ultra-Fine dataset (Choi et al., 2018), >80% of entity labels are with <5 instances, and more seriously 25% of labels never appear in the training data. However, training data can provide very limited direct information for these labels, and therefore previous methods commonly fail to recognize these long-tailed labels.



(a) Extrinsic dependency (b) Intrinsic dependency

(c) Label reasoning process

Figure 1: Examples of deductive reasoning based on the extrinsic dependency and inductive reasoning based on the intrinsic dependency, where the labels *person*, *theorist* and *commander* are deducted respectively and the label *scientist* is inducted from the attributes {expert, scholar}.

*Corresponding authors.

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 4611–4622
November 7–11, 2021.

Fortunately, the implicitly entailed label dependencies in the data provide critical knowledge to tackle the above challenges. Specifically, the dependencies between labels exist extrinsically or intrinsically. On the one hand, the extrinsic dependencies reflect the *direct* connections between labels, which partially appear in the form of label hierarchy and co-occurrence. For example, in Figure 1(a) the labels *person*, *musician*, *composer* are with extrinsic dependencies because they form a three-level taxonomy. Furthermore, *singer* and *composer* are also with extrinsic dependency because they often co-occur with each other. On the other hand, the intrinsic dependencies entail the *indirect* connections between labels through their underlying attributes. For the example in Figure 1(b), label *theorist* and *scientist* share the same underlying attribute of `scholar`. Such intrinsic dependencies provide an effective way to tackle the long tail labels, because many long tail labels are actually composed by non-long tail attributes which can be summarized from non-long tail labels.

To this end, this paper proposes *Label Reasoning Network (LRN)*, which uniformly models, learns and reasons both extrinsic and intrinsic label dependencies without given any predefined label structures. Specifically, LRN utilizes an auto-regressive network to conduct deductive reasoning and a bipartite attribute graph to conduct inductive reasoning between labels. Both of these two kinds of mechanisms are jointly applied to sequentially generate fine-grained labels in an end-to-end, sequence-to-set manner. Figure 1(c) shows several examples. To capture extrinsic dependencies, LRN introduces deductive reasoning (i.e., draw a conclusion based on premises) between labels, and formulates it using an auto-regressive network to predict labels based on both the context and previous labels. For example, given previously-generated label *person* of the mention *they*, as well as the context *they theorize*, LRN will deduce its new label *theorist* based on the extrinsic dependency between *person* and *theorist* derived from data. For intrinsic dependencies, LRN introduces inductive reasoning (i.e., gather generalized information to a conclusion), and utilizes a bipartite attribute graph to reason labels based on current activated attributes of previous labels. For example, if the attributes {`expert`, `scholar`} have been activated, LRN will induce a new label *scientist* based on the attribute-label relations. Consequently, by decomposing labels into attributes and associating long tail labels with frequent labels, LRN can also effectively resolve the long tail label problem by leveraging their non-long tail attributes. Through jointly leveraging the extrinsic and intrinsic dependencies via deductive and inductive reasoning, LRN can effectively handle the massive label set of FET.

Generally, our main contributions are:

- We propose *Label Reasoning Network*, which uniformly models, automatically learns and effectively reasons the complex dependencies between labels in an end-to-end manner.
- To capture extrinsic dependencies, LRN utilizes deductive reasoning to sequentially reason labels via an auto-regressive network. In this way, extrinsic dependencies are discovered and exploited without predefined label structures.
- To capture intrinsic dependencies, LRN utilizes inductive reasoning to reason labels via a bipartite attribute graph. By decomposing labels into attributes and associating long-tailed labels with frequent attributes, LRN can effectively reason long-tailed and even zero-shot labels.

We conduct experiments on standard Ultra-Fine (Choi et al., 2018) and OntoNotes (Gillick et al., 2014) dataset. Experiments show that our method achieves new state-of-the-art performance: a 13% overall F1 improvement and a 44% F1 improvement in the ultra-fine granularity.[1]

2 Related Work

One main challenge for FET is how to exploit complex label dependencies in the large-scale label set. Previous studies typically use predefined label hierarchy and co-occurrence structures estimated from data to enhance the models. To this end, Ren et al. (2016a); Xu and Barbosa (2018); Wu et al. (2019); Chen et al. (2020) design new loss function to exploit label hierarchies. Abhishek et al. (2017) enhance the label representation by sharing parameters. Shimaoka et al. (2017); Murty et al. (2018); López and Strube (2020) embed labels into a high-dimension or a new space. And the studies exploit co-occurrence structures including limiting the label range during label set prediction (Rabinovich and Klein, 2017), enriching the label representation by introducing associated labels (Xiong et al.,

[1]Our source codes are openly available at https://github.com/loriqing/Label-Reasoning-Network

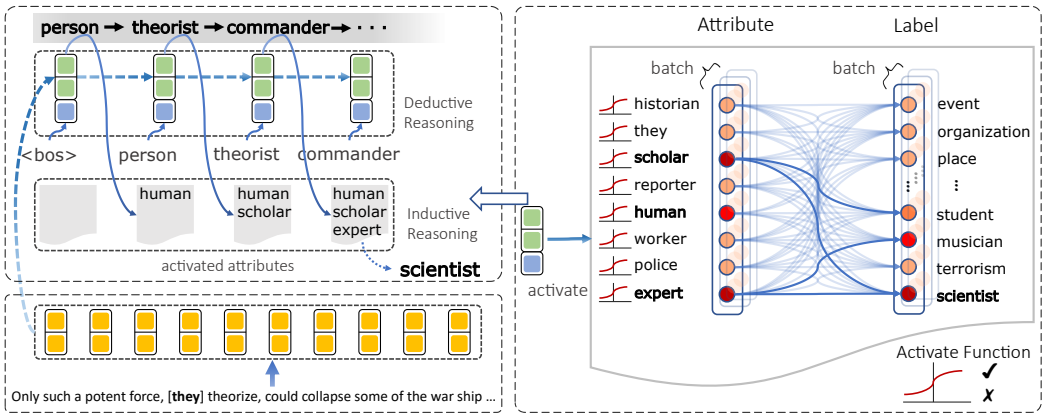


Figure 2: Overview of the process for LRN which contains an encoder, a deductive reasoning-based decoder and an inductive reasoning-based decoder. The figure shows: at step 1, the label *person* is predicted by deductive reasoning, and the attribute `human` is activated; at step 3, the label *scientist* is generated by inductive reasoning.

2019), or requiring latent label representation to reconstruct the co-occurrence structure (Lin and Ji, 2019). However, these methods require predefined label structures or statistics from training data, and therefore is difficult to be extended to new entity types or domains.

The ultra fine-grained label set also leads to data bottleneck and the long tail problem. In recent years, some previous approaches try to tackle this problem by introducing zero/few-shot learning methods (Ma et al., 2016; Huang et al., 2016; Zhou et al., 2018; Yuan and Downey, 2018; Obeidat et al., 2019; Zhang et al., 2020b; Ren et al., 2020), or using data augmentation with denosing strategies (Ren et al., 2016b; Onoe and Durrett, 2019; Zhang et al., 2020a; Ali et al., 2020) or utilizing external knowledge (Corro et al., 2015; Dai et al., 2019) to introduce more external knowledge.

In this paper, we propose Label Reasoning Network, which is significantly different from previous methods because 1) by introducing deductive reasoning, LRN can capture extrinsic dependencies between labels in an end-to-end manner without predefined structures; 2) by introducing inductive reasoning, LRN can leverage intrinsic dependencies to predict long tail labels; 3) Through the sequence-to-set framework, LRN can consider two kinds of label dependencies simultaneously to jointly reason frequent and long tail labels.

3 Label Reasoning Network for FET

Figure 2 illustrates the framework of *Label Reasoning Network*. First, we encode entity mentions through a context-sensitive encoder, then sequentially generate entity labels via two label reasoning mechanisms: deductive reasoning for exploiting extrinsic dependencies and inductive reasoning for exploiting intrinsic dependencies. In our Seq2Set framework, the label dependency knowledge can be effectively modeled in the parameters of LRN, automatically learned from training data, and naturally exploited during the sequential label decoding process. In the following we describe these components in detail.

3.1 Encoding

For encoding, we form the input instance $\mathcal{X}$ as "[CLS], x_1, ..., $[E_1]$, m_1, ..., m_k, $[E_2]$, ..., x_n" where $[E_1]$, $[E_2]$ are entity markers, m is mention word and x is context word. We then feed $\mathcal{X}$ to BERT and obtain the source hidden state $\mathcal{H} = \{h_1, ..., h_n\}$. Finally, the hidden vector of [CLS] token is used as sentence embedding $\boldsymbol{g}$.

3.2 Deductive Reasoning for Extrinsic Dependencies

This section describes how to capture extrinsic dependencies for label prediction via a deductive reasoning mechanism. To this end, the deductive reasoning-based decoder sequentially generates labels based on both context and previous labels, e.g., "*for his books*" + *person* → *writer* and "*record an album*" + *person* → *musician*. In this way, a label is decoded by considering both context-based prediction and previous labels-based prediction.

Concretely, we utilize a LSTM-based autoregressive network as decoder and obtain the hidden state of decoder $\mathcal{S} = \{s_0, ..., s_k\}$, where k is the number of predicted labels. We first initialize s_0 using sentence embedding $\boldsymbol{g}$, then at each time

step, two attention mechanisms – contextual attention and premise attention, are designed to capture context and label information for next prediction.

Contextual Attention is used to capture the context evidence for label prediction. For example, the context "*they theorize*" provides rich information for *theorist* label. Specifically, at each time step t, contextual attention identifies relevant context by assigning a weight α_{ti} to each $\boldsymbol{h}_i$ in the source hidden state $\mathcal{H}$:

$$e_{ti} = \boldsymbol{v}_c^T tanh(\boldsymbol{W}_c \boldsymbol{s}_t + \boldsymbol{U}_c \boldsymbol{h}_i) \quad (1)$$

$$\alpha_{ti} = \frac{exp(e_{ti})}{\sum_{i=1}^{n} exp(e_{ti})} \quad (2)$$

where $\boldsymbol{W}_c$, $\boldsymbol{U}_c$, $\boldsymbol{v}_c$ are weight parameters and $\boldsymbol{s}_t$ is the hidden state of decoder at time step t. Then the context representation $\boldsymbol{c}_t$ is obtained by:

$$\boldsymbol{c}_t = \sum_{i=1}^{n} \alpha_{ti} \boldsymbol{h}_i \quad (3)$$

Premise Attention exploits the dependencies between labels for next label prediction. For example, if *person* has been generated, its hyponym label *theorist* will be highly likely to be generated in context "*they theorize*". Concretely, at each time step t, premise attention captures the dependencies to previous labels by assigning a weight α_{tj} to each $\boldsymbol{s}_j$ of previous hidden states of decoder $\mathcal{S}_{<t}$:

$$e_{tj} = \boldsymbol{v}_p^T tanh(\boldsymbol{W}_p \boldsymbol{s}_t + \boldsymbol{U}_p \boldsymbol{s}_j) \quad (4)$$

$$\alpha_{tj} = \frac{exp(e_{tj})}{\sum_{j=0}^{t-1} exp(e_{tj})} \quad (5)$$

where $\boldsymbol{W}_p$, $\boldsymbol{U}_p$, $\boldsymbol{v}_p$ are weight parameters. Then the previous label information $\boldsymbol{u}_t$ is obtained by:

$$\boldsymbol{u}_t = \sum_{j=0}^{t-1} \alpha_{tj} \boldsymbol{s}_j \quad (6)$$

Label Prediction. Given the context representation $\boldsymbol{c}_t$ and the previous label information $\boldsymbol{u}_t$, we use $\boldsymbol{m}_t = [\boldsymbol{c}_t + \boldsymbol{g}; \boldsymbol{u}_t + \boldsymbol{s}_t]$ as input, and calculate the probability distribution over label set L:

$$\boldsymbol{s}_t = \text{LSTM}(\boldsymbol{s}_{t-1}, \boldsymbol{W}_b \boldsymbol{y}_{t-1}) \quad (7)$$

$$\boldsymbol{o}_t = \boldsymbol{W}_o \boldsymbol{m}_t \quad (8)$$

$$\boldsymbol{y}_t = softmax(\boldsymbol{o}_t + \boldsymbol{I}_t) \quad (9)$$

where $\boldsymbol{W}_o$ and $\boldsymbol{W}_b$ are weight parameters and we use the mask vector $\boldsymbol{I}_t \in \mathbb{R}^{L+1}$ (Yang et al., 2018) to prevent duplicate predictions.

$$(\boldsymbol{I}_t)_i = \begin{cases} -\inf & , l_i \in \mathcal{Y}^*_{t-1} \\ 1 & , \text{otherwise} \end{cases} \quad (10)$$

where $\mathcal{Y}^*_{t-1}$ is the predicted labels before step t and l_i is the i^{th} label in label set L. The label with maximum value in $\boldsymbol{y}_t$ is generated and used as the input for the next time step until $[EOS]$ is generated.

3.3 Inductive Reasoning for Intrinsic Dependencies

Deductive reasoning can effectively capture extrinsic dependencies. However, labels can also have intrinsic dependencies if they share attributes, e.g., *theorist* and *scientist* shares `scholar` attribute. To leverage intrinsic dependencies, LRN conducts inductive reasoning by associating labels to attributes via a bipartite attribute graph. A label will be generated if most of its attributes are activated. Instead of heuristically setting the number of attributes to be activated, we select labels based on their overall activation score from all attributes. By capturing such label-attribute relations, many long tail labels can be effectively predicted because they are usually related to non-long tail attributes.

To this end, we first design a bipartite attribute graph to represent attribute-label relations. Based on the bipartite attribute graph, at each time step, attributes will be activated based on the hidden state of decoder, and new labels will be inducted by reasoning over the activated attributes. For example, in Figure 2 the predicted labels *person*, *theorist* and *commander* will correspondingly activate the attributes `human`, `scholar` and `expert`, and then the *scientist* label will be activated via inductive reasoning based on these attributes.

Bipartite Attribute Graph (BAG). BAG $\mathcal{G} = \{V, E\}$ is designed to capture the relations between attributes and labels. Specifically, nodes V contain attribute nodes V_a and label nodes V_l, and edges E only exist between attributes nodes and labels nodes, with the edge weight indicating the attribute-label relatedness. Attributes are represented using natural language words in BAG. Figure 2 shows a BAG where V_a contains words {`scholar`, `expert`, `historian`, ...}, V_l are all entity labels in label set L, containing {*student*, *musician*, *scientist*, ...}

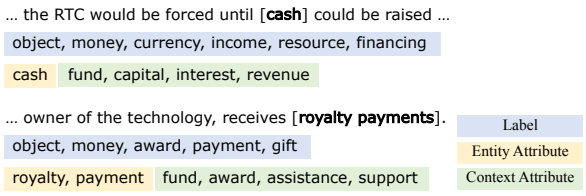


Figure 3: Examples of attributes.

BAG Construction. Because there are many labels and many attributes, we dynamically build a local BAG during the decoding for each instance. In this way the BAG is very compact and the computation is very efficient (Zupan et al., 1999). In local BAG, we collect attributes in two ways: (1) We mask the entity mention in the sentence, and predict the [MASK] token using masked language model (this paper uses BERT-base-uncased), and the non-stop words whose prediction scores greater than a confidence threshold θ_c will be used as attributes — we denote them as context attributes; Since PLM usually predicts high-frequency words, the attributes are usually not long-tailed, which facilitates modeling dependencies between head and tail labels. This mask-prediction strategy is also used in Xin et al. (2018), for collecting additional semantic evidence of entity labels. (2) We directly segment the entity mention into words using Stanza[2], and all non-stop words are used as attributes — we denote them as entity attributes. Figure 3 shows several attribute examples. Given attributes, we compute the attribute-label relatedness (i.e. E in $\mathcal{G}$) using the cosine similarity between their GloVe embeddings (Pennington et al., 2014).

Reasoning over BAG. At each time step, we activate attributes in BAG by calculating their similarities to the current hidden state of decoder $\boldsymbol{s}_t$. For the i^{th} attribute node ${V_a}^{(i)}$, its activation score is:

$$score_{V_a}^{(i)} = ReLU(sim(\boldsymbol{W}_s \boldsymbol{s}_t, \boldsymbol{W}_a {V_a}^{(i)}) \quad (11)$$

where $\boldsymbol{W}_s$ is the weight parameter, $\boldsymbol{W}_a$ is the attribute embedding (i.e., word embedding of attribute words). We use cosine distance to measure similarity and employ ReLU to activate attributes. Then we induce new labels by reasoning over the activated attributes as:

$$score_{V_l}^{(j)} = \sum_{i=1}^{n_a} score_{V_a}^{(i)} E_{ij} \quad (12)$$

where n_a is the number of attributes, ${V_l}^{(j)}$ is the j_{th} label nodes and E_{ij} is the weight between them. Finally a label will be generated if its activation score is greater than a similarity threshold θ_s.

Note that our inductive reasoning and deductive reasoning are jointly modeled in the same decoder, i.e., they share the same decoder hidden state but with different label prediction process. Once deductive reasoning-based decoder generates *[EOS]*, the label prediction stops. Finally, we combine the predicted labels of both deductive reasoning and inductive reasoning as the final FET results.

4 Learning

In FET, each instance is represented as $\{\mathcal{X}, \mathcal{Y}\}$ where $\mathcal{X}$ is "[CLS], x_1, ..., [E_1], m_1, ..., m_k, [E_2],..., x_n" and $\mathcal{Y} = \{y_1, ..., y_m\}$ is the golden labels. To learn our model, we design two losses: set prediction loss for deductive reasoning-based decoding and BAG loss for inductive reasoning-based decoding.

Set Prediction Loss. In FET, cross entropy loss is not appropriate because the prediction results is a label set, i.e., $\{y_1^*, y_2^*, y_3^*\}$ and $\{y_3^*, y_2^*, y_1^*\}$ should have the same loss. Therefore we measure the similarity of two label set using the bipartite matching loss (Sui et al., 2020). Given the golden label set $\mathcal{Y} = \{y_1, ..., y_m\}$ and generated label set $\mathcal{Y}^* = \{y_1^*, ..., y_m^*\}$, the matching loss $\mathcal{L}(ij)_S$ of y_i and y_j^* is calculated by 13, then we use the Hungarian Algorithm (Kuhn, 1955) to get the specific order of golden label set as $\widetilde{\mathcal{Y}} = \{\widetilde{y}_1, ..., \widetilde{y}_m\}$ to obtain minimum matching loss $\mathcal{L}_S$:

$$\mathcal{L}(ij)_S = \text{CE}(y_i, y_j^*) \quad (13)$$

$$\mathcal{L}_S = \text{CE}(\widetilde{\mathcal{Y}}, \mathcal{Y}^*) \quad (14)$$

where CE is cross-entropy.

BAG Loss. To make the model activate labels correctly, we add a supervisory loss to the bipartite attribute graph to active correct labels:

$$\mathcal{L}_A = -\sum_{j=1}^{|L|} score_{V_l}^{(j)} * y_j \quad (15)$$

$$y_j = \begin{cases} 1 & , v_j \in \mathcal{Y} \\ -1 & , v_j \notin \mathcal{Y} \end{cases} \quad (16)$$

Final Loss. The final loss is a combination of set loss and BAG loss:

$$\mathcal{L} = \mathcal{L}_S + \lambda \mathcal{L}_A \quad (17)$$

[2]https://pypi.org/project/stanza/

where λ is the relative weight of these two losses[3].

5 Experiments

5.1 Settings

Datasets We conduct experiments on two standard fine-grained entity typing datasets[4]: Ultra-Fine as primary dataset and OntoNotes as complementary dataset. Ultra-Fine contains 6K manually-annotated examples, 2519 categories, and 5.4 labels per sample on average. Followed Choi et al. (2018) we use the same 2K/2K/2K train/dev/test splits and evaluate using macro precision, recall and F-score. Original OntoNotes dataset (Gillick et al., 2014) contains 25K/2K/9K train/dev/test data, 89 categories and 2.7 labels per sample on average. And Choi et al. (2018) offers an augmentation training data with 2.3 labels per sample on average. We evaluate on both versions using the standard metrics: accuracy, macro F-score and micro F-score.

Baselines For Ultra-Fine dataset, we compare with following baselines: Onoe and Durrett (2019) which offers two multi-classifiers using BERT and ELMo as encoder respectively, Choi et al. (2018) which is a multi-classifier using GloVe+LSTM as encoder, Xiong et al. (2019) which is a multi-classifier using GloVe+LSTM as encoder and exploits label co-occurrence via introducing associated labels to enrich the label representation, López and Strube (2020) which is a hyperbolic multi-classifier using GloVe. For OntoNotes dataset, in addition to the baselines for Ultra-Fine, we also compare with Wang et al. (2020) which offers a multi-classifier using BERT as encoder, Lin and Ji (2019) which offers a multi-classifier using ELMo as encoder and exploits label co-occurrence via requiring the latent representation to reconstruct the co-occurrence association and Chen et al. (2020) which offers a multi-classifier using ELMo as encoder and exploits label hierarchy via designing a hierarchy-aware loss function.

Implementation We use BERT-Base(uncased) (Devlin et al., 2019) as encoder, Adam optimizer (Kingma and Ba, 2015) with learning rate of BERT as 5e-5 and of other parameters as 1e-3. The batch size is 32, encoder hidden size is 768, the decoder hidden size is 868 and label embedding size is 100, the dropout rate of decoder is 0.6. The confidence threshold θ_c and the similarity threshold θ_s both are optimized on dev set and set as 0.1 and 0.2 respectively. We use the GloVe embedding (Pennington et al., 2014) to represent the nodes of BAG and fix it while training.

[3]In our auxiliary experiments, we find that its impact is minor, so this paper empirically sets it to 1.

[4]Released in https://github.com/uwnlp/open_type

Model	P	R	F1
without label dependency			
*Choi et al. (2018)	47.1	24.2	32.0
*ELMo(Onoe and Durrett, 2019)	51.5	33.0	40.2
BERT(Onoe and Durrett, 2019)	51.6	33.0	40.2
BERT[in-house]	55.9	33.0	41.5
with label dependency			
*LABELGCN (Xiong et al., 2019)	50.3	29.2	36.9
LRN w/o IR	**61.2**	33.5	43.3
LRN	54.5	**38.9**	**45.4**

Table 1: Macro P/R/F1 results on Ultra-Fine test set. * means using augmented data. "without label dependency" methods formulated FET as multi-label classification without considering associations between labels. "with label dependency" methods leveraged associations between labels explicitly or implicitly.

5.2 Overall Results

Table 1 shows the main results of all baselines and our method in two settings: LRN is the full model and LRN w/o IR is the model without inductive reasoning. For fair comparisons, we implement a baseline with same settings of LRN but replace the decoder with a multi-classifier same as Choi et al. (2018) — BERT[in-house]. We can see that:

1) *By performing label reasoning, LRN can effectively resolve the fine-grained entity typing problem.* Compared with previous methods, our method achieves state-of-the-art performance with a F1 improvement from 40.2 to 45.4 on test set. This verified the necessity for exploiting label dependencies for FET and the effectiveness of our two label reasoning mechanisms. We believe this is because label reasoning can help FET by making the learning more data-efficient (i.e., labels can share knowledge) and the prediction of labels global coherent.

2) *Both deductive reasoning and inductive reasoning are useful for fine-grained label prediction.* Compared with BERT[in-house], LRN w/o IR can achieve 4.3% F1 improvement by exploiting extrinsic dependencies via deductive reasoning. LRN can further improve F1 from 43.3 to 45.4 by exploiting intrinsic dependencies via inductive reasoning. We believe this is because deductive reasoning and inductive reasoning are two fundamental but different mechanisms, therefore, modeling them simultaneously will better leverage label dependencies to

Model	Total			General			Fine			Ultra-Fine		
	P	R	F	P	R	F	P	R	F	P	R	F
*Choi et al. (2018)	48.1	23.2	31.3	60.3	61.6	61.0	40.4	38.4	39.4	42.8	8.8	14.6
†LABELGCN (Xiong et al., 2019)	49.3	28.1	35.8	66.2	68.8	67.5	43.9	40.7	42.2	42.4	14.2	21.3
HY Large (López and Strube, 2020)	43.4	34.2	38.2	61.4	73.9	67.1	35.7	46.6	40.4	36.5	19.9	25.7
*ELMo (Onoe and Durrett, 2019)	50.7	33.1	40.1	66.9	**80.7**	73.2	41.7	46.2	43.8	45.6	17.4	25.2
BERT (Onoe and Durrett, 2019)	51.6	32.8	40.1	67.4	80.6	73.4	41.6	54.7	47.3	46.3	15.6	23.4
BERT[in-house]	54.1	32.1	40.3	68.8	79.2	73.6	43.8	**57.4**	49.7	**50.7**	14.6	22.6
LRN w/o IR	**60.7**	32.5	42.3	**79.3**	75.5	**77.4**	**59.6**	44.8	51.2	45.7	18.7	26.5
LRN	53.7	**38.6**	**44.9**	77.8	76.4	77.1	55.8	50.6	**53.0**	43.4	**26.0**	**32.5**

Table 2: Macro P/R/F1 of each label granularity on Ultra-Fine dev set, and long tail labels are mostly in the ultra-fine layer. * means using augmented data. † We adapt the results from López and Strube (2020).

Model	Total			General			Fine			Ultra-Fine		
	P	R	F	P	R	F	P	R	F	P	R	F
HY XLarge (López and Strube, 2020)	/	/	/	/	/	69.1	/	/	39.7	/	/	26.1
BERT[in-house]	55.9	33.0	41.5	69.7	**81.6**	75.2	43.7	**56.0**	49.1	**53.5**	15.5	24.0
LRN w/o IR	**61.2**	33.5	43.3	**78.3**	76.7	**77.5**	**61.6**	44.1	51.4	47.8	19.9	28.1
LRN	54.5	**38.9**	**45.4**	77.4	76.7	77.1	58.4	50.4	**54.1**	43.5	**26.4**	**32.8**

Table 3: Macro P/R/F1 of different label granularity on Ultra-Fine test set.

Number of	Category	Prediction	Shot=0			Shot=1			Shot=2		
			Correct	Predicted	Prec.	Correct	Predicted	Prec.	Correct	Predicted	Prec.
BERT[in-house]	293	5683	0	0	/	1	1	100.0%	9	66	13.6%
LRN w/o IR	330	5740	0	0	/	1	3	33.3%	15	28	53.6%
LRN	997	7808	110	218	50.5%	67	252	26.6%	94	276	34.1%

Table 4: Performance of the zero-shot, shot=1 and shot=2 label prediction. "Category" means how many kinds of types are predicted. "Prediction" means how many labels are generated.

predict labels.

3) *Seq2Set is an effective framework to model, learn and exploit label dependencies in an end-to-end manner.* Compared with LABELGCN (Xiong et al., 2019) which heuristically exploits label co-occurrence structure, LRN can achieve a significant performance improvement. We believe this is because neural networks have strong ability for representing and learning label dependencies. And the end-to-end manner makes LRN can easily generalize to new scenarios.

5.3 Effect on Long Tail Labels

As described above, another advantage of our method is it can resolve the long tail problem by decomposing long tail labels to common attributes and modeling label dependencies between head and tail labels. Because the finer the label granularity, the more likely it to be a long tail label, we report the performance of each label granularity on dev set and test set same as previous works in Table 2 and Table 3. Moreover, we report the performance of the labels with shot$\leq$2 in Table 4. Based on these results, we find that:

1) *LRN can effectively resolve the long tail label problem.* Compared to BERT[in-house], LRN can significantly improve the F-score of ultra-fine granularity labels by 44% (22.6 $\rightarrow$ 32.5) and recall more fine-grained labels (14.6 $\rightarrow$ 26.0).

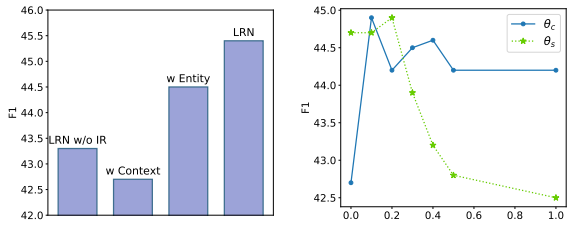
2) *Both deductive reasoning and inductive reasoning are helpful for long tail label prediction, but with different underlying mechanisms: deductive reasoning exploits the extrinsic dependencies between labels, but inductive reasoning exploits the intrinsic dependencies between labels.* LRN w/o IR cannot predict zero-shot labels because it resolves long tail labels by relating head labels with long tail labels, therefore it cannot predict unseen labels. By contrast, LRN can predict zero-shot labels via inductive reasoning because it can decompose labels into attributes. Furthermore, we found LRN w/o IR has higher precision for few-shot (shot=2) labels than BERT and LRN, we believe this is because inductive reasoning focuses on recalling more labels, which inevitably introduce some incorrect labels.

5.4 Detailed Analysis

Effect of Components To evaluate the effect of different components, we report the ablation results in Table 5. We can see that: (1) Set prediction loss is effective: replacing it with cross-entropy loss will lead to a significant decrease. (2) Both context

Model	Dev			Test		
	P	R	F	P	R	F
LRN	53.7	38.6	44.9	54.5	38.9	45.4
-PreAtt	53.1	39.3	45.2	52.6	39.5	45.1
-PreAtt-ConAtt	56.3	36.3	44.2	56.4	36.5	44.3
-SetLoss	46.8	40.7	43.5	47.8	40.7	44.0
LRN w/o IR	60.7	32.5	42.3	61.2	33.5	43.3
-PreAtt	54.5	34.2	42.1	55.1	35.0	42.8
-PreAtt-ConAtt	55.2	32.9	41.3	56.2	34.3	42.6
-SetLoss	46.0	37.6	41.4	46.6	37.5	41.6

Table 5: Ablation results on Ultra-Fine dataset: PreAtt denotes premise attention, ConAtt denotes contextual attention, and -SetLoss denotes replacing set prediction loss with cross-entropy loss.



(a) (b)

Figure 4: (a) Ablation experiments of context attributes and entity attributes on Ultra-Fine dataset. (b) Performances of different confidence threshold θ_c and similarity threshold θ_s on dev set.

and premise attention mechanisms are important for Seq2Set generation.

Effect of Attributes Set To explore the impact of entity attributes and context attributes in BAG, Figure 4(a) shows the results of different attributes configurations. We can see that: both attributes are useful, the context attribute has high coverage and may be noisy, while the entity attribute is opposite. However when introducing both of them, the information in entity attributes might help the context attributes to disambiguate them. This is similar to the effectiveness of contextual information in word sense disambiguation. As a result, these two kinds of attributes can complement each other. And Figure 4(b) shows the performance on different thresholds, and we optimize confidence threshold $\theta_c = 0.1$ and similarity threshold $\theta_s = 0.2$ on dev set. Notice that θ_s is the threshold of activating labels and when $\theta_s = 1$, it is equivalent to LRN w/o IR .

Results of OntoNotes To verify the generality of our method, we further conduct experiments on OntoNotes and report results of with and without augmentation data in Table 6. To embed labels in OntoNotes, we use the embedding of the last word of a label, e.g., */person/artist/director* is represented using embedding of *director*.

Encoder	Model	Acc	MaF	MiF
with augmentation				
HYPER	López and Strube (2020)	47.4	75.8	69.4
LSTM	Choi et al. (2018)	59.5	76.8	71.8
	Xiong et al. (2019)	59.6	77.8	72.2
ELMo	*Onoe and Durrett (2019)	64.9	84.5	79.2
	(Lin and Ji, 2019)	63.8	82.9	77.3
BERT	Wang et al. (2020)	61.1	81.8	76.3
	BERT [in-house]	62.2	83.4	78.8
	LRN w/o IR	**66.1**	**84.8**	**80.1**
	LRN	64.5	84.5	79.3
without augmentation				
ELMo	*Onoe and Durrett (2019)	42.7	72.7	66.7
	Chen et al. (2020)	**58.7**	73.0	68.1
BERT	Onoe and Durrett (2019)	51.8	76.6	69.1
	BERT[in-house]	51.5	76.6	69.7
	LRN w/o IR	55.3	77.3	70.4
	LRN	56.6	**77.6**	**71.8**

Table 6: Results on OntoNotes test set. Augmentation is the augmented data created by (Choi et al., 2018) which contains 800K instances and therefore there're little few-shot labels in this setting. And * indicates using additional features to enhance the label representation.

We can see that: 1) LRN still achieves the best performance on both settings, which verified the robustness of our method. 2) Compared with Ultra-Fine, our method achieves a smaller improvement on OntoNotes. We found this is mainly because: First, OntoNotes has weaker label dependencies for its label set is smaller (89 vs 2519 for Ultra-Fine) and most of its labels are coarse-grained. Secondly, most labels in OntoNotes are frequent labels with many training instances, therefore the long tail label problem is not serious. This also explains why LRN w/o IR can achieve better performance than LRN in the setting of with augmentation data: the more the training instance, the less need for long tail prediction.

5.5 Case Study

To intuitively present the learned label dependencies, Figure 5 shows the label co-occurrence matrices of different models' predictions and ground truth, we can see that both LRN and LRN w/o IR can accurately learn label dependencies. Figure 6 shows some prediction cases and demonstrates that deductive and inductive reasoning have quite different underlying mechanisms and predict quite different labels.

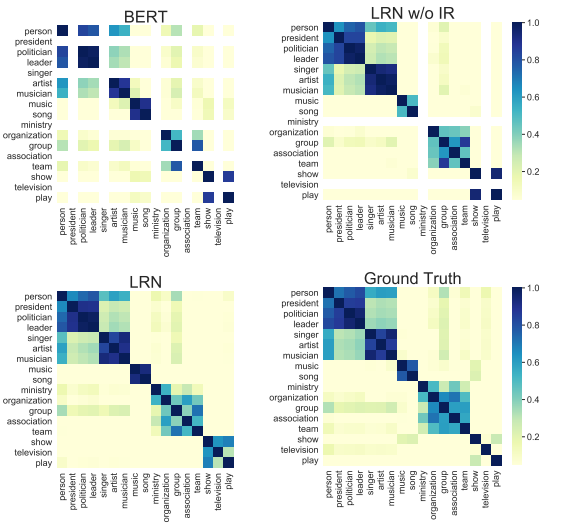


Figure 5: Heat map of co-occurrence matrices of different models' prediction and ground truth. LRN w/o IR and LRN learn very similar co-occurrence matrices to Ground Truth.

[**Estadio Jose Duarte de Paiva**], is a multi-use stadium located in Brazil .		
ANSWER	place structure building stadium arena	
BERT	place structure	
LRN	place → structure → stadium	
	{arena}	
As [**a concert artist and composer**], he teaches students throughout …		
ANSWER	person adult male man professional performer artist musician entertainer creator teacher composer	
BERT	person musician artist composer	
LRN	person → musician → artist → creator	Deductive Reasoning
	{professor, teacher, composer}	Inductive Reasoning

Figure 6: Cases of prediction results.

6 Conclusions

This paper proposes *Label Reasoning Network*, which uniformly models, learns and reasons complex label dependencies in a sequence-to-set, end-to-end manner. LRN designs two label reasoning mechanisms for effective decoding – deductive reasoning to exploit extrinsic dependencies and inductive reasoning to exploit intrinsic dependencies. Experiments show that LRN can effectively cope with the massive label set on FET. And because our method uses no predefined structures, it can be easily generalized to new datasets and applied to other multi-classification tasks.

7 Acknowledgments

This work is supported by the National Key Research and Development Program of China (No. 2020AAA0106400), the National Natural Science Foundation of China under Grants no.U1936207 and 62106251, Beijing Academy of Artificial Intelligence (BAAI2019QN0502), and in part by the Youth Innovation Promotion Association CAS(2018141).

References

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, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 1: Long Papers*, pages 797–807. Association for Computational Linguistics.

Muhammad Asif Ali, Yifang Sun, Bing Li, and Wei Wang. 2020. Fine-grained named entity typing over distantly supervised data based on refined representations. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7391–7398. AAAI Press.

Tongfei Chen, Yunmo Chen, and Benjamin Van Durme. 2020. Hierarchical entity typing via multi-level learning to rank. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8465–8475. 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, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 87–96. Association for Computational Linguistics.

Luciano Del Corro, Abdalghani Abujabal, Rainer Gemulla, and Gerhard Weikum. 2015. FINET: context-aware fine-grained named entity typing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 868–878. The Association for Computational Linguistics.

Hongliang Dai, Donghong Du, Xin Li, and Yangqiu Song. 2019. Improving fine-grained entity typing with entity linking. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 6209–6214. 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

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

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). *CoRR*, abs/1603.03112.

Sanjeev Karn, Ulli Waltinger, and Hinrich Schütze. 2017. End-to-end trainable attentive decoder for hierarchical entity classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers*, pages 752–758. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Harold W Kuhn. 1955. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97.

Hongyu Lin, Yaojie Lu, Xianpei Han, and Le Sun. 2019a. Sequence-to-nuggets: Nested entity mention detection via anchor-region networks. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5182–5192. Association for Computational Linguistics.

Hongyu Lin, Yaojie Lu, Xianpei Han, Le Sun, Bin Dong, and Shanshan Jiang. 2019b. Gazetteer-enhanced attentive neural networks for named entity recognition. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 6231–6236. Association for Computational Linguistics.

Hongyu Lin, Yaojie Lu, Jialong Tang, Xianpei Han, Le Sun, Zhicheng Wei, and Nicholas Jing Yuan. 2020. A rigorous study on named entity recognition: Can fine-tuning pretrained model lead to the promised land? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 7291–7300. Association for Computational Linguistics.

Ying Lin and Heng Ji. 2019. An attentive fine-grained entity typing model with latent type representation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 6196–6201. Association for Computational Linguistics.

Federico López and Michael Strube. 2020. A fully hyperbolic neural model for hierarchical multi-class classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020, Online Event, 16-20 November 2020*, pages 460–475. Association for Computational Linguistics.

Yukun Ma, Erik Cambria, and Sa Gao. 2016. Label embedding for zero-shot fine-grained named entity typing. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 171–180. ACL.

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, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 97–109. Association for Computational Linguistics.

Rasha Obeidat, Xiaoli Z. 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 807–814. Association for Computational Linguistics.

Yasumasa Onoe and Greg Durrett. 2019. Learning to denoise distantly-labeled data for entity typing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2407–2417. Association for Computational Linguistics.

Yasumasa Onoe and Greg Durrett. 2020. Fine-grained entity typing for domain independent entity linking. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8576–8583. AAAI Press.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543. ACL.

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, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 2: Short Papers*, pages 330–334. 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. 2016a. AFET: automatic fine-grained entity typing by hierarchical partial-label embedding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1369–1378. The Association for Computational Linguistics.

Xiang Ren, Wenqi He, Meng Qu, Clare R. Voss, Heng Ji, and Jiawei Han. 2016b. Label noise reduction in entity typing by heterogeneous partial-label embedding. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1825–1834. ACM.

Yankun Ren, Jianbin Lin, and Jun Zhou. 2020. Neural zero-shot fine-grained entity typing. In *Companion of The 2020 Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, pages 846–847. ACM / IW3C2.

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, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 1: Long Papers*, pages 1271–1280. Association for Computational Linguistics.

Dianbo Sui, Yubo Chen, Kang Liu, Jun Zhao, Xiangrong Zeng, and Shengping Liu. 2020. Joint entity and relation extraction with set prediction networks. *CoRR*, abs/2011.01675.

Yanping Wang, Xin Xin, and Ping Guo. 2020. An empirical study of pre-trained embedding on ultra-fine entity typing. In *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2656–2661. IEEE.

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 *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 5264–5270. ijcai.org.

Ji Xin, Hao Zhu, Xu Han, Zhiyuan Liu, and Maosong Sun. 2018. Put it back: Entity typing with language model enhancement. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 993–998. Association for Computational Linguistics.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 773–784. Association for Computational Linguistics.

Peng Xu and Denilson Barbosa. 2018. Neural fine-grained entity type classification with hierarchy-aware loss. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 16–25. Association for Computational Linguistics.

Yadollah Yaghoobzadeh, Heike Adel, and Hinrich Schütze. 2017. Noise mitigation for neural entity typing and relation extraction. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 1: Long Papers*, pages 1183–1194. Association for Computational Linguistics.

Pengcheng Yang, Xu Sun, Wei Li, Shuming Ma, Wei Wu, and Houfeng Wang. 2018. SGM: sequence generation model for multi-label classification. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 3915–3926. Association for Computational Linguistics.

Semih Yavuz, Izzeddin Gur, Yu Su, Mudhakar Srivatsa, and Xifeng Yan. 2016. Improving semantic parsing via answer type inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 149–159. The Association for Computational Linguistics.

Zheng Yuan and Doug Downey. 2018. Otyper: A neural architecture for open named entity typing. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 6037–6044. AAAI Press.

Haoyu Zhang, Dingkun Long, Guangwei Xu, Muhua Zhu, Pengjun Xie, Fei Huang, and Ji Wang. 2020a. Learning with noise: Improving distantly-supervised fine-grained entity typing via automatic relabeling. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3808–3815. ijcai.org.

Tao Zhang, Congying Xia, Chun-Ta Lu, and Philip S. Yu. 2020b. MZET: memory augmented zero-shot fine-grained named entity typing. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 77–87. International Committee on Computational Linguistics.

Wenkai Zhang, Hongyu Lin, Xianpei Han, and Le Sun. 2021a. De-biasing distantly supervised named entity recognition via causal intervention. 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 4803–4813, Online. Association for Computational Linguistics.

Wenkai Zhang, Hongyu Lin, Xianpei Han, Le Sun, Huidan Liu, Zhicheng Wei, and Nicholas Jing Yuan. 2021b. Denoising distantly supervised named entity recognition via a hypergeometric probabilistic model. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 14481–14488. AAAI Press.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: enhanced language representation with informative entities. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1441–1451. Association for 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, Brussels, Belgium, October 31 - November 4, 2018*, pages 2065–2076. Association for Computational Linguistics.

Blaz Zupan, Marko Bohanec, Janez Demsar, and Ivan Bratko. 1999. Learning by discovering concept hierarchies. *Artif. Intell.*, 109(1-2):211–242.