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

Code: https://github.com/loriqing/Label-Reasoning-Network

——EMNLP2021

2022. 03. 17

Reported by ChangJiang Hu

Introduction

What is Fine-grained entity typing(FET)

Given a candidate entity(mention) and it is context, predict a set of possible categories(Type)

Context: "They were arrested by FBI agents."
Mention: FBI agents
Type:{organization, administration, force, agent, police}.

Case: Jack robs Mike, Jack is eventually caught.

NER:Jack:{person},Mike:{person}

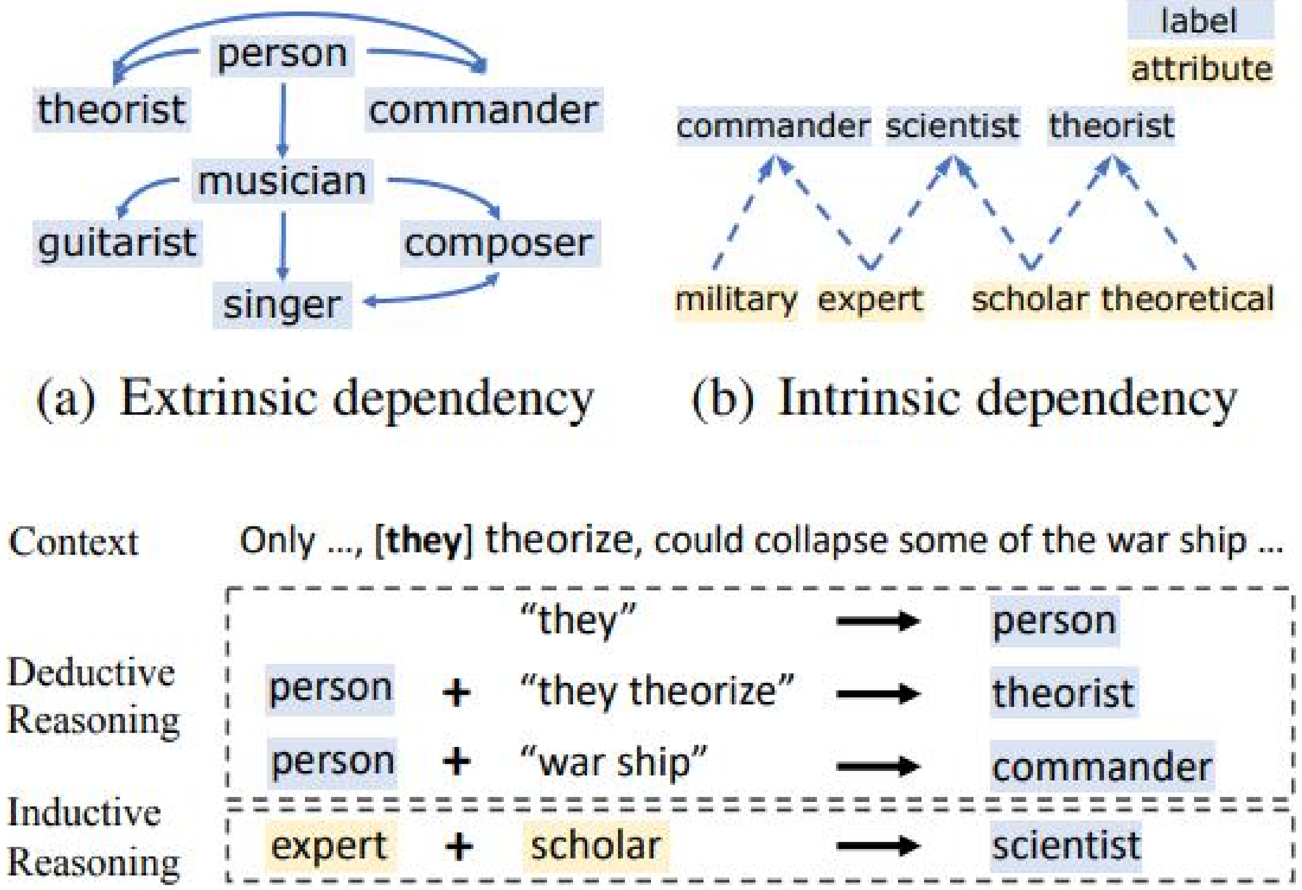
FET:Jack:{person, criminal},Mike:{person, victim}

By providing fine-grained semantic labels, FET is critical for entity recognition and can benefit many NLP tasks, such as relation extraction ,entity linking and question answering.

Introduction

First Due to the massive label set, it is impossible to independently recognize each entity label without considering their dependencies

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



(c) Label reasoning process

Method

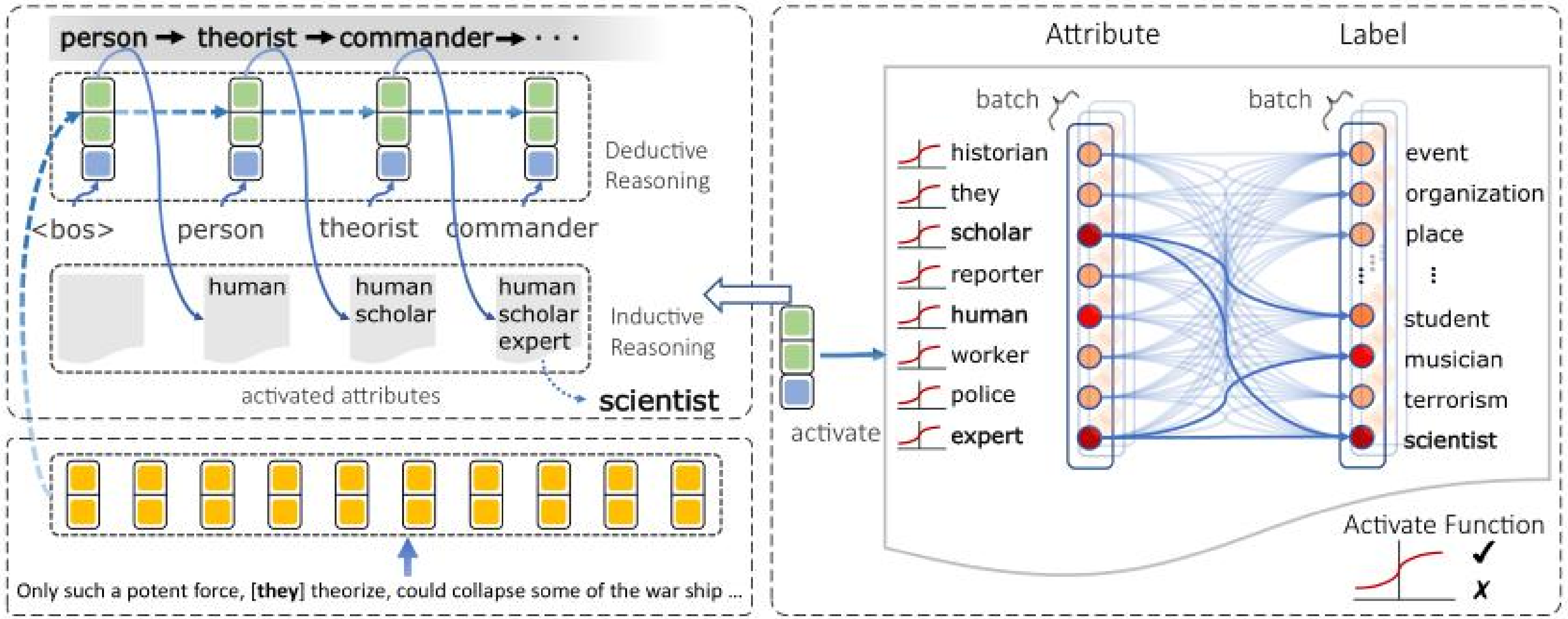
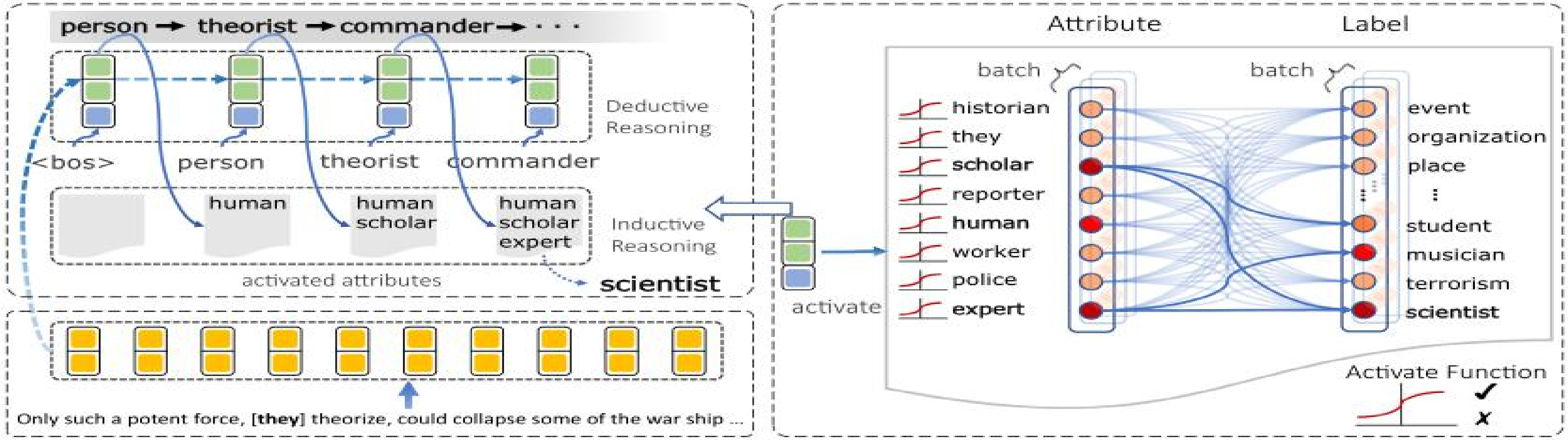


Figure 2: Overview of the process for LRN which contains an encoder, a deductive reasoning-based decoder and an inductive reasoning-based decoder. The figure shows: at step 1, the label *person* is predicted by deductive reasoning, and the attribute human is activated; at step 3, the label *scientist*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

For encoding, we form the input instance 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 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 **g**

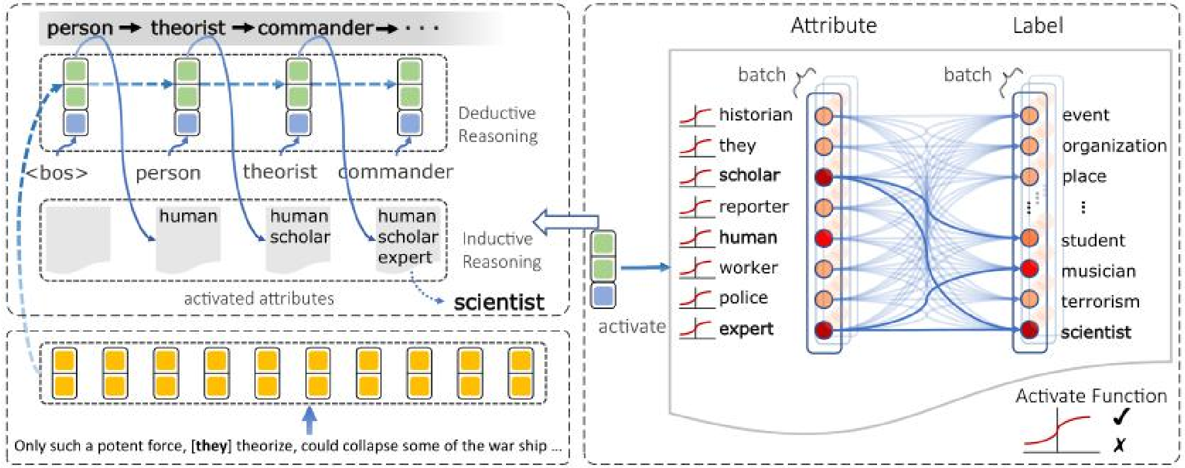
Introduction

Deductive Reasoning for Extrinsic Dependencies



Concretely, we utilize a LSTM-based auto-regressive network as decoder and obtain the hidden state of decoder $S = \{s_0, \ldots, s_k\}$, where k is the number of predicted labels. We first initialize s_0 using sentence embedding **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.

Method



$$s_t = \text{LSTM}(s_{t-1}, W_b y_{t-1}) \quad (7) \qquad \mathcal{S} = \{s_0, \ldots, s_k\}$$

Contextual Attention

$$e_{ti} = v_c^T tanh(W_c s_t + U_c h_i) \quad (1)$$

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

$$c_t = \sum_{i=1}^{n} \alpha_{ti} h_i \quad (3)$$

Premise Attention

$$e_{tj} = v_p^T tanh(W_p s_t + U_p s_j) \quad (4)$$

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

$$u_t = \sum_{j=0}^{t-1} \alpha_{tj} s_j \quad (6)$$

Label Prediction

$$m_t = [c_t + g; u_t + s_t]$$

$$o_t = W_o m_t \quad (8)$$

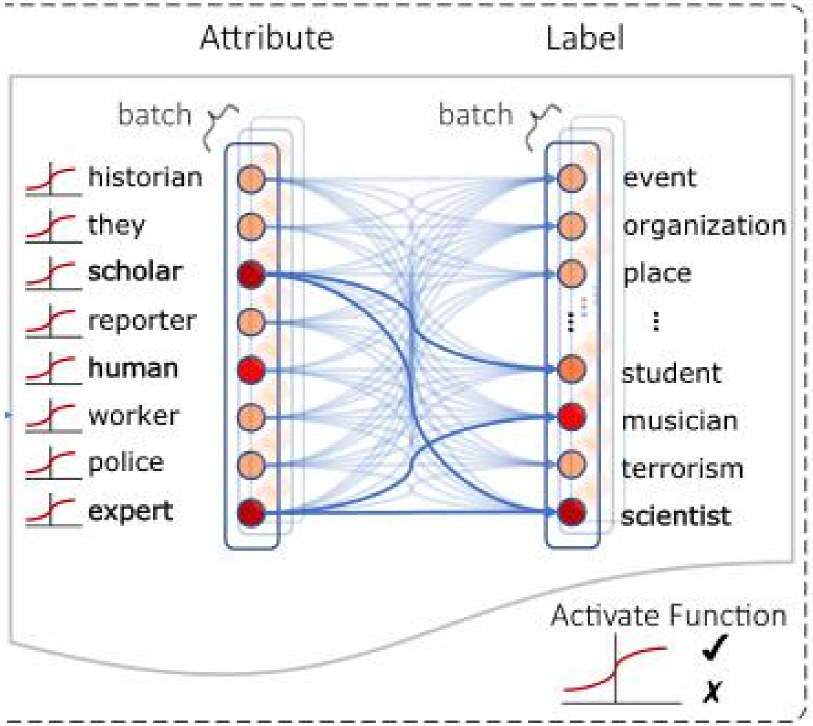
$$y_t = softmax(o_t + I_t) \quad (9)$$

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

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 y_t is generated and used as the input for the next time step until $[EOS]$ is generated.

Method

Inductive Reasoning for Intrinsic Dependencies



BAG Construction

BAG **g** ={V, E}

V contain attribute nodes V_a and label nodes V_l

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 ,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.
 (2) We directly segment the entity mention into words using Stanza2, 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 **g**) using the cosine similarity between their GloVe embeddings.

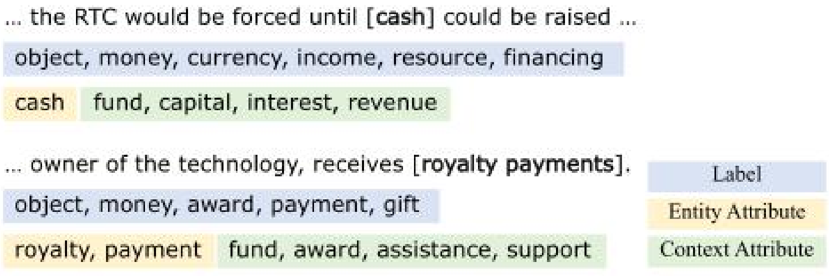
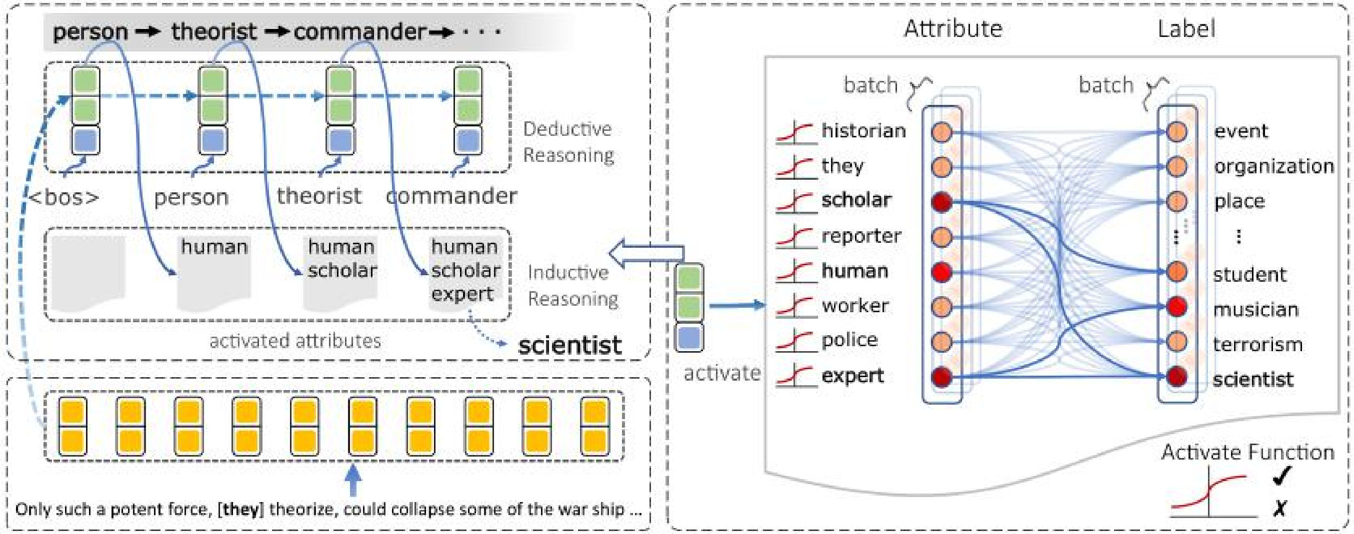


Figure 3: Examples of attributes.

Method

Reasoning over BAG



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

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

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.

Method

Learning

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 $\mathcal{L}_S$:

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

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

where CE is cross-entropy.

Joint Entity and Relation Extraction with Set Prediction Networks

Dianbo Sui♡♠ Yubo Chen♡ Kang Liu♡♠ Jun Zhao♡♠
Xiangrong Zeng◇ Shengping Liu◇
♡ National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China
♠ University of Chinese Academy of Sciences, Beijing, China
◇ Beijing Unisound Information Technology Co., Ltd, Beijing, China
{dianbo.sui, yubo.chen, kliu, jzhao}@nlpr.ia.ac.cn, {zengxiangrong, liushengping}@unisound.com

$$\begin{aligned}\mathcal{L}(\mathbf{Y}, \hat{\mathbf{Y}}) = \sum_{i=1}^{m} \{ & -\log \mathbf{p}^{r}_{\pi^*(i)}(r_i) \\ & + \mathbb{1}_{\{r_i \neq \varnothing\}}[-\log \mathbf{p}^{s-start}_{\pi^*(i)}(s_i^{start}) \\ & - \log \mathbf{p}^{s-end}_{\pi^*(i)}(s_i^{end}) \\ & - \log \mathbf{p}^{o-start}_{\pi^*(i)}(o_i^{start}) \\ & - \log \mathbf{p}^{o-end}_{\pi^*(i)}(o_i^{end})]\}\end{aligned}$$

Experiments

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

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

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

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

$$\mathcal{L} = \mathcal{L}_S + \lambda \mathcal{L}_A \tag{17}$$

Experiments

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.

Experiments

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.

Experiments

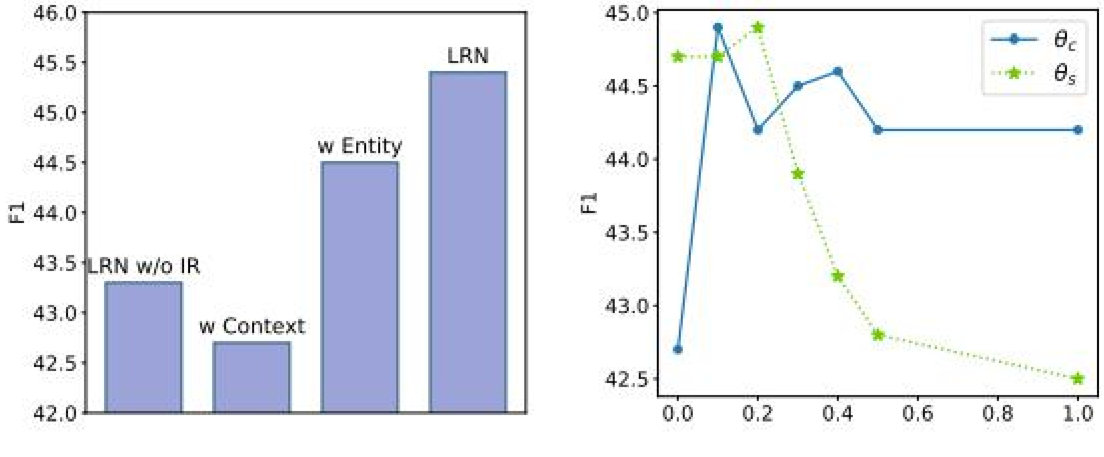
Number of	Category	Prediction	Shot=0			Shot=1			Shot=2		
			Correct	Predicted	Prec.	Correct	Predicted	Prec.	Correct	Predicted	Prec.
BERT[in-house]	293	5683	0	0	/	1	1	100.0%	9	66	13.6%
LRN w/o IR	330	5740	0	0	/	1	3	33.3%	15	28	53.6%
LRN	997	7808	110	218	50.5%	67	252	26.6%	94	276	34.1%

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.

Experiments

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.

Experiments

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.

Thank you!