

Modeling Label Correlations for Ultra-Fine Entity Typing with Neural Pairwise Conditional Random Field

Chengyue Jiang[†], Yong Jiang^{☆*}, Weiqi Wu, Pengjun Xie[☆], Kewei Tu^{*}

[☆]DAMO Academy, Alibaba Group, China

{jiangchy1997, kewei.tu.nlp}@gmail.com; wuwq1022@foxmail.com;
{yongjiang.jy, chengchen.xpj}@alibaba-inc.com

Abstract

Ultra-fine entity typing (UFET) aims to predict a wide range of type phrases that correctly describe the categories of a given entity mention in a sentence. Most recent works infer each entity type independently, ignoring the correlations between types, e.g., when an entity is inferred as a *president*, it should also be a *politician* and a *leader*. To this end, we use an undirected graphical model called pairwise conditional random field (PCRF) to formulate the UFET problem, in which the type variables are not only unarily influenced by the input but also pairwise relate to all the other type variables. We use various modern backbones for entity typing to compute unary potentials, and derive pairwise potentials from type phrase representations that both capture prior semantic information and facilitate accelerated inference. We use mean-field variational inference for efficient type inference on very large type sets and unfold it as a neural network module to enable end-to-end training. Experiments on UFET show that the Neural-PCRF consistently outperforms its backbones with little cost and results in a competitive performance against cross-encoder based SOTA while being *thousands of times* faster. We also find Neural-PCRF effective on a widely used fine-grained entity typing dataset with a smaller type set. We pack Neural-PCRF as a network module that can be plugged onto multi-label type classifiers with ease and release it in github.com/modelscope/adaseq/examples/NPCRF.

1 Introduction

Entity typing assigns semantic types to entities mentioned in the text. The extracted type information has a wide range of applications. It acts as a primitive for information extraction (Yang and Zhou, 2010) and spoken language understanding

(Coucke et al., 2018), and assists in more complicated tasks such as machine reading comprehension (Joshi et al., 2017) and semantic parsing (Yavuz et al., 2016). During its long history of development, the granularity of the type set for entity typing and recognition changes from coarse (sized less than 20) (Tjong Kim Sang and De Meulder, 2003; Hovy et al., 2006), to fine (sized around 100) (Weischedel and Brunstein, 2005; Ling and Weld, 2012; Gillick et al., 2014; Ding et al., 2021b), to ultra-fine and free-form (sized 10k) (Choi et al., 2018). The expansion of the type set reveals the diversity of real-world entity categories and the importance of a finer-grained understanding of entities in applications (Choi et al., 2018).

The increasing number of types results in difficulties in predicting correct types, so a better understanding of the entity types and their correlations is needed. Most previous works solve an N -type entity typing problem as N independent binary classifications (Ling and Weld, 2012; Gillick et al., 2014; Choi et al., 2018; Onoe and Durrett, 2019; Ding et al., 2021a; Pan et al., 2022; Li et al., 2022). However, types are highly correlated and hence they should be predicted in a joint manner. As an example, when ‘Joe Biden’ is inferred as a *president*, it should also be inferred as a *politician*, but not a *science fiction*. Type correlation is partially specified in fine-grained entity typing (FET) datasets by a two or three-level type hierarchy, and is commonly utilized by a hierarchy-aware objective function (Ren et al., 2016; Jin et al., 2019; Xu and Barbosa, 2018). However, type hierarchies cannot encode type relationships beyond strict containment, such as similarity and mutual exclusion (Onoe et al., 2021), and could be noisily defined (Wu et al., 2019) or even unavailable (in ultra-fine entity typing (UFET)). Many recent works handle these problems by embedding types and mentions into special spaces such as the hyperbolic space (López and Strube, 2020) or box space (Onoe et al.,

[†]This work was done during Chengyue Jiang’s internship at DAMO Academy, Alibaba Group.

*Yong Jiang and Kewei Tu are corresponding authors.

2021) that can be trained to latently encode type correlations without a hierarchy. Although these methods are expressive for modeling type correlations, they are constrained by these special spaces and thus incapable of being combined with modern entity typing backbones such as prompt learning (Ding et al., 2021a) and cross-encoder (Li et al., 2022), and cannot integrate prior type semantics.

In this paper, we present an efficient method that expressively models type correlations while being backbone-agnostic. We formulate the UFET and FET problems under a classical undirected graphical model (UGM) (Koller and Friedman, 2009) called pairwise conditional random field (PCRF) (Ghamrawi and McCallum, 2005). In PCRF, types are binary variables that are not only unarily influenced by the input but also pairwise relate to all the other type variables. We formulate the unary potentials using the type logits provided by any modern backbone such as prompt learning. To compose the pairwise potentials sized $O(4N^2)$ (N is the number of types, 10k for UFET), we use matrix decomposition, which is efficient and able to utilize prior type semantic knowledge from word embeddings (Pennington et al., 2014). Exact inference on such a large and dense UGM is intractable, and therefore we apply mean-field variational inference (MFVI) for approximate inference. Inspired by Zheng et al. (2015), we unfold the MFVI as a recurrent neural network module and connect it with the unary potential backbone to enable end-to-end training and inference. We call our method Neural-PCRF (NPCRF). Experiments on UFET show that NPCRF consistently outperforms the backbones with negligible additional cost, and results in a strong performance against cross-encoder based SOTA models while being *thousands of times* faster. We also found NPCRF effective on a widely used fine-grained entity typing dataset with a smaller type set. We pack NPCRF as a network module that can be plugged onto multi-label type classifiers with ease and release it in github.com/modelscope/adaseq/examples/NPCRF.

2 Related Work

2.1 Fine-grained and Ultra-Fine Entity Typing

Fine-grained (Gillick et al., 2014; Ling and Weld, 2012; Weischedel and Brunstein, 2005) (FET) and ultra-fine (Choi et al., 2018) entity typing (UFET) share common challenges. Researchers put lots of

effort into better utilizing the distantly and manually labeled training data, including rule-based denoising (Gillick et al., 2014), multi-round automatic denoising (Onoe and Durrett, 2019; Pan et al., 2022; Dai et al., 2021), and partial label embedding (Ren et al., 2016). Some recent works introduce better entity typing backbones compared with commonly used multi-label classifiers (Gillick et al., 2014; Choi et al., 2018; Onoe and Durrett, 2019), such as prompt learning (Ding et al., 2021a), cross-encoders with indirect supervision from natural language inference (Li et al., 2022) and using a graph neural network to enhance mention representations by sibling mentions for FET (Chen et al., 2022). Another line of work concentrates on modeling the given or latent type correlations in entity typing, which is more relevant to our work. Ren et al. (2016); Jin et al. (2019); Xu and Barbosa (2018) leverage a given type hierarchy through a hierarchy-aware loss for FET. However, in UFET, the type hierarchy is not given. Therefore various special type embeddings are designed to learn the latent type correlation during training. Xiong et al. (2019) utilize graph embedding of a label co-occurrence graph counted from training data, López and Strube (2020) embeds entity mentions, contexts and labels into the hyperbolic space that can latently encode tree structures, and Onoe et al. (2021) embeds those into the box space that encodes label dependency through the topology of hyperrectangles. Unlike them, we treat types as random variables under an undirected probabilistic graphical model, and only require the type logits, therefore our neural PCRF is backbone-agnostic. Liu et al. (2021) propose a label reasoning network (LRN) that generates types in an auto-regressive manner, while our method decodes types in parallel and can be combined with prompt-based backbones which LRN cannot do.

2.2 Pairwise Conditional Random Field

Pairwise conditional random field (Ghamrawi and McCallum, 2005) is a classical undirected graphical model proposed for modeling label correlations. In the deep learning era, PCRF was first found effective when combined with a convolutional neural network (CNN) (Fukushima and Miyake, 1982) for semantic segmentation in computer vision (Zheng et al., 2015; Chandra and Kokkinos, 2016; Arnab et al., 2016; Shen et al., 2017; Lê-Huu and Alahari, 2021a), in which it was used to encourage

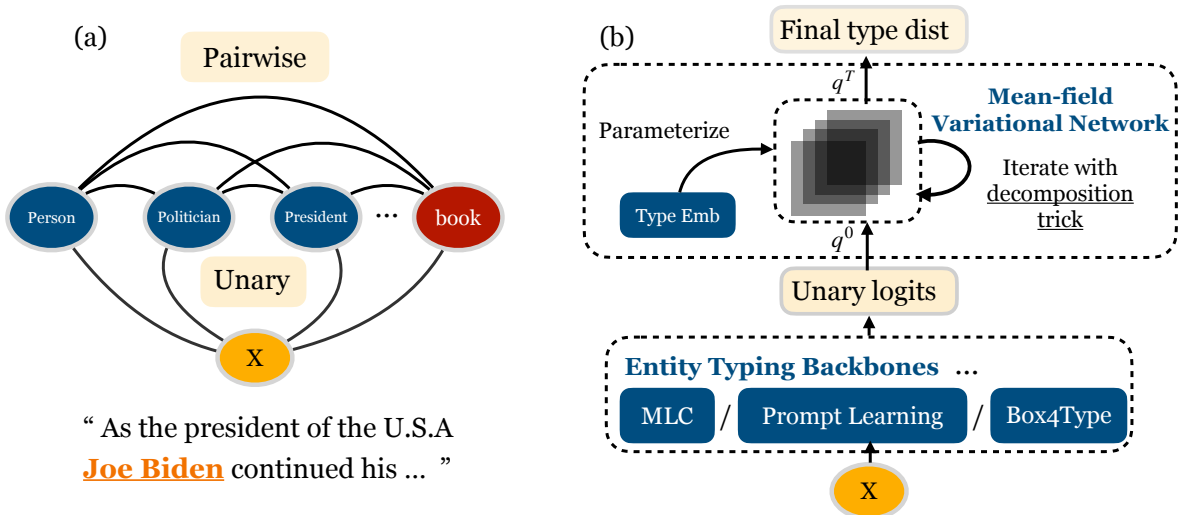


Figure 1: (a) Our pairwise CRF (PCRF) of ultra-fine entity typing. X is the entity mention and its context. (b) The neural network architecture corresponding to the Neural-PCRF.

adjacent pixels being segmented together. In contrast to its popularity in computer vision, PCRF is much less explored in natural language processing. Ghamrawi and McCallum (2005) and Wang et al. (2017) apply PCRF on an n-gram feature based non-neural sentence classifier with up to 203 classes. Besides the difference in tasks, numbers of classes and backbones, our method is different from theirs in two main aspects: (1) We unfold mean-field approximate inference of PCRF as a recurrent neural network module for efficient end-to-end training and inference, while they use the ‘supported inference method’ (Ghamrawi and McCallum, 2005) which is intractable for large type sets and incompatible with neural backbones (2) We design the parameterization of the pairwise potential function based on the technique of matrix decomposition to accelerate training and inference, and encode type semantics which is important for entity typing (Li et al., 2022) and sentence classification (Mueller et al., 2022). Our parameterization also conforms to intrinsic properties of pairwise potentials (explained in Sec. 3.4). Hu et al. (2020) investigates different potential function variants for linear-chain CRFs for sequence labeling, while we design pairwise potential functions for PCRF.

3 Method

3.1 Problem Definition

Entity typing datasets consist of M entity mentions m_i with their corresponding context sentences c_i : $\mathcal{D} = \{(m_1, c_1), \dots, (m_M, c_M)\}$, and a type set \mathcal{Y}

of size N . The task of entity typing is to predict the types \mathbf{y}_i^p of the entity mention m_i in the given context c_i , where \mathbf{y}_i^p is a subset of the type set. The number of gold types $|\mathbf{y}_i^g|$ could be larger than one in most entity typing datasets. The average number of gold types per instance $\text{avg}(|\mathbf{y}_i^g|)$ and the size of the type set $|\mathcal{Y}|$ vary in different datasets.

3.2 PCRF for Entity Typing

We first describe our pairwise conditional random field for the entity typing (Ghamrawi and McCallum, 2005), as shown in Fig. 1(a). x denotes a data point $(m, c) \in \mathcal{D}$. $Y_j \in \{0, 1\}$ denotes the binary random variable for the j -th type in type set \mathcal{Y} . The type variables $Y_{1:N}$ are unarily connected to the input, model how likely a type is given the input, and pairwise connect a type variable with all other type variables to model type correlations. Let $\mathbf{y} \in Y_1 \times Y_2 \times \dots \times Y_N$ be an assignment of all the type variables. The probability of \mathbf{y} given a data point x_i under a conditional random field can be factorized as:

$$p(\mathbf{y}|x) = \frac{1}{Z} \exp \left(\sum_j \theta_u(y_j; x) + \sum_{j < k} \theta_p(y_j, y_k) \right)$$

where θ_u, θ_p are scoring functions for unary and pairwise edges, and Z is the partition function for normalization.

3.3 Unary Potential Function

As illustrated in Figure 1(b), we parameterize the log unary potential function $\theta_u(y_j; x)$ using modern entity typing backbones based on pretrained

potential	encode	property
$\Theta_{11}(j, k) = \theta_p(y_j = 1, y_k = 1)$	co-occurrence	$\Theta_{11} = \Theta_{11}^T$
$\Theta_{00}(j, k) = \theta_p(y_j = 0, y_k = 0)$	co-absence	$\Theta_{00} = \Theta_{00}^T$
$\Theta_{10}(j, k) = \theta_p(y_j = 1, y_k = 0)$	co-exclusion	$\Theta_{10} = \Theta_{01}^T$
$\Theta_{01}(j, k) = \theta_p(y_j = 0, y_k = 1)$	co-exclusion	$\Theta_{01} = \Theta_{10}^T$

Table 1: Properties of the four log potential matrices.

language models (PLM)¹. We introduce two of the backbones below.

Multi-label Classifier (MLC) Given a data point (m, c) , we formulate the input as:

$$[\text{CLS}] c [\text{SEP}] m [\text{SEP}]$$

and feed it into RoBERTa-large (Liu et al., 2019) to obtain the [CLS] embedding \mathbf{h} . Then we obtain the unary type scores as follows.

$$\begin{aligned} \theta_u(y_j = 1; x) &= \text{Linear}(\mathbf{h})[j] \\ \theta_u(y_j = 0; x) &= 0 \end{aligned}$$

Prompt Learning (PL) We adopt prompt learning for entity typing (Ding et al., 2021a; Pan et al., 2022) as our second backbone. We feed:

$$[\text{CLS}] c [\text{P}_1] m [\text{P}_2] [\text{P}_3] [\text{MASK}] [\text{SEP}]$$

into Bert-cased (Devlin et al., 2018), where $[\text{P}_1], [\text{P}_2], [\text{P}_3]$ are trainable soft prompt tokens. We use a verbalizer \mathcal{V} to map types to subwords, \mathcal{V}_y denotes the subwords corresponding to type $y \in \mathcal{Y}$, e.g., $\mathcal{V}_{\text{living_thing}} = \{\text{‘living’}, \text{‘thing’}\}$. We obtain the unary score of choosing type y_j using the averaged masked language model logits of \mathcal{V}_{y_j} :

$$\theta_u(y_j = 1; x) = \sum_{w \in \mathcal{V}_{y_j}} s([\text{MASK}] = w) / |\mathcal{V}_{y_j}|$$

Similarly, we set $\theta(y_j = 0; x) = 0$.

3.4 Pairwise Potential Function

Pairwise log potential $\theta_p(y_j, y_k)$ encoding type correlations should naturally satisfy the properties shown in Table 1. Specifically, Θ_{11} and Θ_{00} are symmetric matrices and encode co-occurrence and co-absence respectively, while $\Theta_{10} = \Theta_{01}^T$ and are asymmetric (e.g., “not a person but a president” \neq “not a president but a person”). Directly parameterizing these 4 potential matrices (Ghamrawi and McCallum, 2005; Wang et al., 2017) ignores these

¹We use PLMs from <https://huggingface.co/>

intrinsic properties and results in an unbearable number of model parameters for datasets with a large type set (e.g., 400M parameters for 10331-type UFET, which is more than Bert-large).

To tackle these problems, we parameterize the pairwise potential based on matrix rank decomposition, i.e., we represent each $N \times N$ matrix as the product of two $N \times R$ matrices, where R is the number of ranks. Crucially, we use pretrained word embedding to derive the two $N \times R$ matrices, so that they can encode type semantics. Specifically, we obtain a type embedding matrix $\mathbf{E} \in \mathbb{R}^{N \times 300}$ based on 300-dimension GloVe embedding (Pennington et al., 2014). For a type phrase consisting of multiple words (e.g., “living_thing”), we take the average of the word embeddings. We then transform \mathbf{E} into two embedding spaces encoding “occurrence” and “absence” respectively using two feed-forward networks.

$$\begin{aligned} \text{FFN}(\mathbf{E}) &= \mathbf{W}_2(\tanh(\text{dropout}(\mathbf{W}_1 \mathbf{E}))) \\ \mathbf{E}_0 &= \text{FFN}_1(\mathbf{E}) \in \mathbb{R}^{N \times R} \\ \mathbf{E}_1 &= \text{FFN}_2(\mathbf{E}) \in \mathbb{R}^{N \times R} \end{aligned} \quad (1)$$

To enforce the intrinsic properties of the four matrices, we parameterize them as follows:

$$\begin{aligned} \Theta_{00} &= \mathbf{E}_0 \mathbf{E}_0^T & \Theta_{11} &= \mathbf{E}_1 \mathbf{E}_1^T \\ \Theta_{01} &= -\mathbf{E}_0 \mathbf{E}_1^T & \Theta_{10} &= -\mathbf{E}_1 \mathbf{E}_0^T \end{aligned} \quad (2)$$

The negative signs in defining Θ_{01}, Θ_{10} ensure that they encode co-exclusion, not similarity. As we will show in the next subsection, we do not need to actually recover these large matrices during inference, leading to lower computational complexity.

3.5 Mean-field Variational Inference

We aim to infer the best type set given the input.

$$\mathbf{y}^p = \underset{\mathbf{y} \in \mathcal{Y}_1 \times \dots \times \mathcal{Y}_N}{\text{argmax}} p(\mathbf{y}|x)$$

The MAP (maximum-a-posteriori) inference over this large and dense graph is NP-hard. We instead use mean-field variational inference (MFVI) to approximately infer the best assignment \mathbf{y}^p . MFVI approximates the true posterior distribution $p(y_j|x)$ by a variational distribution $q(y_j)$ that can be factorized into independent marginals (Wainwright et al., 2008). It iteratively minimizes the KL-divergence between p and q . We initialize the q distribution by unary scores produced by backbones:

$$q^0(y_j) = \text{softmax}(\theta_u(y_j; x))$$

and derive the MFVI iteration as follows:

$$\begin{aligned}
q^{t+1}(y_j = 1) &\propto \exp\left(\theta_u(y_j = 1; x) + \sum_{k \neq j} (\Theta_{11}^{(jk)} \cdot q^t(y_k = 1)) + \Theta_{10}^{(jk)} \cdot q^t(y_k = 0)\right) \\
q^{t+1}(y_j = 0) &\propto \exp\left(\theta_u(y_j = 0; x) + \sum_{k \neq j} (\Theta_{00}^{(jk)} \cdot q^t(y_k = 0)) + \Theta_{01}^{(jk)} \cdot q^t(y_k = 1)\right)
\end{aligned}$$

We rewrite the formulas in the vector form with our parametrization. $\theta_1^u \in \mathbb{R}^N$ denotes the unary score vector of $\theta_u(y_j = 1; x)$ for all j , and we define θ_0^u similarly. $\mathbf{q}_0^t, \mathbf{q}_1^t$ are vectors of the q distributions at the t -th iteration.

$$\begin{aligned}
(\mathbf{q}_0^0[j], \mathbf{q}_1^0[j]) &= \text{softmax}(\theta_0^u[j], \theta_1^u[j]) \\
\mathbf{q}_1^{t+1} &\propto \exp\left(\theta_1^u + \mathbf{E}_1(\mathbf{E}_1^T \mathbf{q}_1^t) - \mathbf{E}_1(\mathbf{E}_0^T \mathbf{q}_0^t)\right) \\
\mathbf{q}_0^{t+1} &\propto \exp\left(\theta_0^u + \mathbf{E}_0(\mathbf{E}_0^T \mathbf{q}_0^t) - \mathbf{E}_0(\mathbf{E}_1^T \mathbf{q}_1^t)\right)
\end{aligned}$$

Note that by following these update formulas, we do not need to recover the $N \times N$ matrices, and hence the time complexity of each iteration is NR . Since the iteration number \mathcal{T} is typically a small constant ($\mathcal{T} < 7$), the computational complexity of the whole MFVI is $O(NR)$. The predicted type set of x_i is obtained by:

$$\mathbf{y}^p = \{y_j \mid \mathbf{q}_1^{\mathcal{T}}[j] > 0.5, y_j \in \mathcal{Y}\}$$

We follow the treatment of MFVI as entropy-regularized Frank-Wolfe (Lê-Huu and Alahari, 2021b) and introduce another hyper-parameter λ to control the step size of each update. Let $\mathbf{q}^t = [\mathbf{q}_0^t; \mathbf{q}_1^t]$,

$$\mathbf{q}^{t+1} = \mathbf{q}^t + \lambda(\mathbf{q}^{t+1} - \mathbf{q}^t)$$

3.6 Unfolding Mean-field Variational Inference as a Recurrent Neural Network

We follow Zheng et al. (2015) and treat the mean-field variational inference procedure with a fixed number of iterations as a recurrent neural network (RNN) parameterized by the pairwise potential functions. As shown in the top part of Figure 1(b), the initial hidden state of the RNN is the type distribution \mathbf{q}^0 produced by the unary logits, and the final hidden states $\mathbf{q}^{\mathcal{T}}$ after \mathcal{T} iterations are used for end-to-end training and inference.

3.7 Training Objective and Optimization

We use the Binary Cross Entropy loss for training:

$$\mathcal{L} = -\frac{1}{N} \sum_{j=1}^N \left(\alpha \cdot y_j^g \log \mathbf{q}_1^{\mathcal{T}}[j] + (1 - y_j^g) \log(1 - \mathbf{q}_1^{\mathcal{T}}[j]) \right)$$

\mathcal{L} is the loss of each instance x , $y_j^g \in \{0, 1\}$ is the gold annotation of x for type y_j . We follow previous works (Choi et al., 2018; Dai et al., 2021) and use α as a weight for the loss of positive types. We train the pretrained language model, label embedding matrix \mathbf{E} and FFNs. We use AdamW (Loshchilov and Hutter, 2018) for optimization.

4 Experiment

4.1 Datasets and Experimental Settings

We mainly evaluate our NPCRF method on the ultra-fine entity typing dataset: UFET (Choi et al., 2018). We also conduct experiments on the augmented version (Choi et al., 2018) of OntoNotes (Gillick et al., 2014) to examine if our method also works for datasets with smaller type sets and fewer gold types per instance. We show the dataset statistics in Table 2. Note that UFET also provides 25M distantly labeled training data extracted by linking to KB and parsing. We follow recent works (Pan et al., 2022; Liu et al., 2021) and only use the manually annotated 2k data for training. We use standard metrics for evaluation: for UFET, we report macro-averaged precision (P), recall (R), and F1; for OntoNotes, we report macro-averaged and micro-averaged F1. We run experiments three times and report the average results.

dataset	$ \mathcal{Y} $	avg($ \mathbf{y}_i^g $)	train/dev/test	hierarchy
UFET	10331	5.4	2k/2k/2k	No
OntoNotes	89	1.5	0.8M/2k/9k	3-layer

Table 2: $|\mathcal{Y}|$ denotes the size of the type set, avg($|\mathbf{y}_i^g|$) denotes the average number of gold types per instance.

4.2 Baseline Methods

MLC-ROBERTA and **PL-BERT** introduced in Sec. 3.3 are natural baselines. We compare their performances with and without NPCRF.

UFET (Choi et al., 2018) A multi-label linear classifier (MLC) with a backbone using BiLSTM, GloVe, and CharCNN to encode the mention.

LDET (Onoe and Durrett, 2019) An MLC with

MODEL	P	R	F1
<i>non-PLM model for reference</i>			
UFET-BiLSTM (Choi et al., 2018) [†]	48.1	23.2	31.3
LABELGCN (Xiong et al., 2019) ^{†◇}	50.3	29.2	36.9
<i>PLM-based models</i>			
LDET (Onoe and Durrett, 2019) [†]	51.5	33.0	40.1
BOX4TYPES (Onoe et al., 2021) ^{†◇}	52.8	38.8	44.8
LRN (Liu et al., 2021) [◇]	54.5	38.9	45.4
MLMET (Dai et al., 2021) [†]	53.6	45.3	49.1
LITE+L (Li et al., 2022)	48.7	45.8	47.2
DFET (Pan et al., 2022)	55.6	44.7	49.5
LITE+NLI+L (Li et al., 2022) [†]	52.4	48.9	50.6
<i>PLM-based models w/ or w/o NPCRF</i>			
MLC-ROBERTA	47.8	40.4	43.8
MLC-ROBERTA w/ NPCRF	48.7	45.5	47.0
LABELGCN-ROBERTA [◇]	51.2	41.0	45.5
LABELGCN-ROBERTA w/ NPCRF	48.7	45.9	47.3
PL-BERT-BASE (Ding et al., 2021a)	57.8	40.7	47.7
PL-BERT-BASE w/ NPCRF	52.1	47.5	49.7
PL-BERT-LARGE	59.3	42.6	49.6
PL-BERT-LARGE w/ NPCRF	55.3	46.7	50.6

Table 3: Macro-averaged UFET result. **LITE+L** is LITE without NLI pretraining, **LITE+L+NLI** is the full LITE model, **LABELGCN-ROBERTA** denotes our implementation of LabelGCN with RoBERTa-large as mention encoder. Methods marked by [†] use either distantly labeled training data or additional pretraining tasks and [◇] marker denotes method focusing on modeling label correlations.

Bert-base-uncased and ELMo (Peters et al., 2018) and trained by 727k examples automatically denoised from the distantly labeled UFET.

LABELGCN (Xiong et al., 2019) An MLC with BiLSTM and multi-head self-attention to encode the mention and context, and a GCN running on a fixed co-occurrence type graph to obtain better type embedding. Type scores are dot-product of mention and type embeddings. For fair comparison, we replace their mention encoder with RoBERTa-large. To our knowledge, LabelGCN cannot be directly combined with prompt learning.

BOX4TYPE (Onoe et al., 2021) use Bert-large as backbone and project mentions and types to the box space for training and inference; trained on the same 727k data (Onoe and Durrett, 2019).

LRN (Liu et al., 2021) generate types using Bert-base and an LSTM decoder in a seq2seq manner, and use 2k manually labeled data for training.

MLMET (Dai et al., 2021) A multi-label linear classifier using Bert-base, first pretrained by the distantly-labeled data augmented by masked word prediction, then finetuned and self-trained on the

2k human-annotated data.

DFET (Pan et al., 2022) A 3-round automatic denoising method for 2k manually labeled data, using **PL-BERT** as backbone.

LITE (Li et al., 2022) Previous SOTA system that formulates entity typing as natural language inference, treating x_i as premise and types as hypothesis, concatenating them and feeding them into RoBERTa-large to score types. LITE models the correlation better between the input and types, and has great zero-shot performance but needs to concatenate x_i with all the N types, resulting in very slow inference. LITE is pretrained on MNLI (Williams et al., 2018) and trained on 2k annotated data, Its authors find that the performance drops when using distantly labeled data.

4.3 UFET Result

As shown in Table 3, our **NPCRF** can be integrated with various entity typing backbones and enhance their performance to SOTA on UFET.

MLC-ROBERTA w/ NPCRF improves the basic **MLC-ROBERTA** backbone by +3.2 F1 score and reaches the best performance among models using the MLC architecture except for **MLMET** which uses millions of distantly labeled data. **LABELGCN** (Xiong et al., 2019) utilizing fixed type co-occurrence information to obtain type embedding is still effective with the replaced **ROBERTA** encoder and improves F1 by +1.7 over **MLC-ROBERTA** while our method produce a further +1.8 F1 improvement, because **NPCRF** models not only co-occurrence (in Θ_{11}), but also semantic co-absence and co-exclusion in our pairwise potentials, and it also explicitly updates these potentials during training rather than using fixed type co-occurrence information.

Prompt-learning based backbones (Ding et al., 2021a; Pan et al., 2022) such as **PL-BERT** are already strong in UFET, and our method further improves **PL-BERT-BASE** by +1.9 F1 and reaches 49.7 F1 which is slightly better than **MLMET** and **DFET** and is the SOTA performance for Bert-base models, on par with performance of Bert-large models. Also worth noting is that our method is trained in a simpler single-round end-to-end manner compared with other competitors requiring multi-round training (Dai et al., 2021) and denoising (Pan et al., 2022) procedure. For prompt-learning powered by Bert-large models, **NPCRF** boosts the performance of **PL-BERT-LARGE** by

MODEL	MA-F1	MI-F1
<i>PLM-based models</i>		
LDET (Onoe and Durrett, 2019) †	84.5	79.2
BOX4TYPES (Onoe et al., 2021) †◇	77.3	70.9
LRN (Liu et al., 2021) ◇	84.8	80.1
MLMET (Dai et al., 2021) †	85.4	80.4
LITE (Li et al., 2022)	86.6	81.4
DFET (Pan et al., 2022)	87.1	81.5
<i>PLM-based models w/ or w/o NPCRF</i>		
MLC-ROBERTA	85.4	80.2
MLC-ROBERTA w/ NPCRF	86.9	81.9

LABELGCN-ROBERTA ◇	85.5	80.2
LABELGCN-ROBERTA w/ NPCRF	86.2	81.3

PL-BERT-BASE (Ding et al., 2021a)	84.8	79.6
PL-BERT-BASE w/ NPCRF	85.2	80.0

Table 4: Results on the augmented OntoNotes datasets. **MA** and **MI** are abbreviations of macro and micro. Methods marked by † use either distantly labeled training data or additional pretraining tasks and the ◇ marker denotes methods modeling label correlations.

1.0 F1, and results in performance on par with the previous SOTA system **LITE+NLI+L** with much faster inference speed (discussed in Sec. 5.5). Note that the improvement is smaller on large models because large models are stronger in inferring types without pairwise potentials from limited and noisy contexts. We also observe that models with **NPCRF** tend to predict more types and therefore have higher recalls and sometimes lower precisions. It is possibly because **NPCRF** can use type dependencies to infer additional types that are not directly supported by the input. We will discuss it in detail in Sec. 5.4.

4.4 FET Result

We present the performance of **NPCRF** on the augmented OntoNotes dataset in Table 4. The results show that **MLC-ROBERTA w/ NPCRF** outperforms **MLC-ROBERTA** by +1.5 in macro-F1 and +1.7 in micro-F1, and reaches competitive performances against a recent SOTA system **DFET** focusing on denoising data. In general, we find our method still effective, but the improvement is less significant compared with UFET, especially for **PL-BERT**. One possible reason is that the average number of types per instance in OntoNotes is 1.5, and therefore the type-type relationship is less important. Another possible reason is that some type semantics are already covered in the prompt-based method through the verbalizer, so our method fails to boost the performance of **PL-BERT**.

MODEL	MA-P	MA-R	MA-F1
PL-Bert w/ NPCRF	52.1	47.5	49.7
- w/o GloVe	48.5	49.1	48.8
- w/o FFN hidden	51.1	47.0	49.0
- w/o FFN	-	-	-
- w/o NPCRF	57.8	40.7	47.7

Table 5: Ablation study on UFET test set, “-” means training fails to converge.

MODEL	MI-P	MI-R	MI-F1
<i>coarse-grained, 9-class, avg(y_i^g) = 0.95</i>			
MLC-ROBERTA	76.5	75.8	76.2
LABELGCN-ROBERTA	77.5	74.7	76.1
MLC-ROBERTA w/ NPCRF	78.0	76.0	77.0
<i>fine-grained, 130-class, avg(y_i^g) = 1.61</i>			
MLC-ROBERTA	62.8	67.3	64.9
LABELGCN-ROBERTA	59.7	69.6	64.3
MLC-ROBERTA w/ NPCRF	62.7	68.8	65.6

Table 6: Performances on coarser granularity.

5 Analysis

5.1 Ablation Study

We show the ablation study in Table 5. It can be seen that: (1) Performance drops when we randomly initialize label embedding E (denoted by **- w/o GloVe**), which indicates that GloVe embedding contains useful type semantics and helps build pairwise type correlations. (2) Performance drops when we remove the hidden layer and the tanh non-linearity in **FFNs** (Eq. 1), showing the benefit of expressive parameterization. (3) When we remove the entire **FFN** layers and use E to parameterize the four matrices in (Eq. 2) model training fails to converge, showing that a reasonable parameterization is important.

5.2 Performance on Coarser Granularity

We evaluate the performance of **NPCRF** on type sets with coarser granularity. UFET (Choi et al., 2018) splits the types into coarse-grained, fine-grained and ultra-fine-grained types. We discard ultra-fine-grained types to create the fine-grained setting (130 types), and further discard the fine-grained types to create the coarse-grained setting (9 types). As shown in Table 6, **NPCRF** still has positive effect in these two settings, while **LABELGCN** does not.

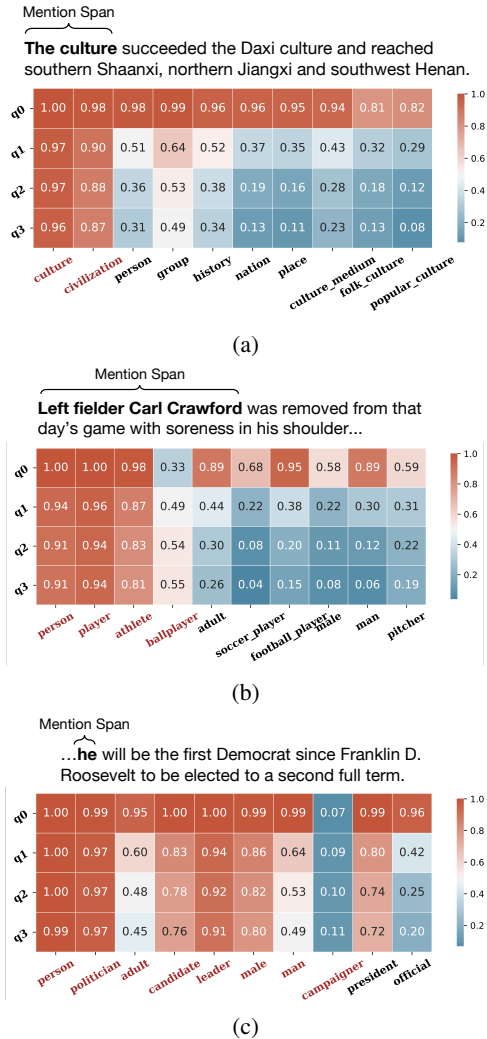


Figure 2: Two MFVI cases. We show, from top to bottom, how the type probabilities q_t^t change with iteration given the mention and its context. The reddish grids (probability > 0.5) indicates the chosen types, and gold types are colored red on the x-axis.

5.3 Model Performance at Each Iteration

We evaluate the per-iteration performance of **PL-BERT w/ NPCRF** on the test set and show the results in Table 7. We obtain the prediction y_t^p of each iteration t by binarizing q_t for each instance: $y_t^p = \{y_j \mid q_1^t[j] > 0.5, y_j \in \mathcal{Y}\}$. The results show that model prediction at iteration 0 (i.e., based solely on unary scores) has a high recall and a very low precision, while **NPCRF** keeps correcting wrong candidates during consecutive iterations to reach a higher F1 score. We show some concrete cases in Sec. 5.4.

5.4 Case Study and Visualization

MFVI Iterations We show the per-iteration predictions of **PL-BERT w/ NPCRF** for two inputs

ITERATION	MA-P	MA-R	MA-F1	$\text{avg}(y_{i,t}^p)$
q^0	15.3	74.5	25.4	27.92
q^1	43.2	53.6	47.8	6.52
q^2	50.0	48.2	49.1	4.95
q^3	52.1	47.4	49.7	4.65

Table 7: Model performance at each iteration, $\text{avg}(|y_{i,t}^p|)$ denotes the average number of predicted types.

in Figure 2. As can be seen, **NPCRF** tends to delete wrong types through iterations, such as “*history*”, “*nation*” in Fig. 2(a) and “*pitcher*”, “*soccer_player*” in Fig. 2(b). This results in higher precision (as shown in Table 7). However, we find **NPCRF** is also capable of increasing the probabilities of some types, e.g., “*ballplayer*” in the second case. **NPCRF** may also erroneously deprecate gold types and results in lower recall. As been shown in 2(c), **NPCRF** wrongly deletes gold types such as “*adult*”, “*man*” while it correctly predicts “*president*” which is not annotated as gold label, and increases the score of “*campaigner*”.

Pairwise Potentials We show the four learned pairwise potentials in Appendix A.1.

5.5 Efficiency

We compare the training and inference efficiency of different methods on the UFET dataset in Table 8. We run all these methods on one Tesla V100 GPU three times, and report the average speed (number of sentences per second) during training and inference. Except for **LRN** which is based on Bert-base-uncased, all the other methods are based on RoBERTa-large. Results show that **NPCRF** (4 iterations, using the best hyper-parameters) is the fastest to model type correlations (compared with **LABELGCN** and **LRN**), it slows down training by 15.7% and inference by only 13.8%. **NPCRF** is much faster than **LITE** which is based on the cross-encoder architecture in inference².

6 Conclusion and Future Work

We propose **NPCRF**, a method that efficiently models type correlation for ultra-fine and fine-grained entity typing, and is applicable to various entity typing backbones. In **NPCRF**, the unary potential is formulated as the type logits of modern UFET backbones, the pairwise potentials are

²Prompt-learning based method has similar time complexity as MLC based method theoretically. We do not compare them because of differences in the PLMs and implementations.

MODEL (sents/sec)	TRAIN	INFERENCE
MLC-ROBERTA	37.04	142.86
MLC-ROBERTA w/ NPCRF	31.25	123.46

LABELGCN-ROBERTA	27.78	90.91
LRN	19.59	20.88

LITE	20.41	0.03

Table 8: Comparison of training and inference speed of different methods in Table 3.

derived from type phrase representations that both capture prior semantic information and facilitate accelerated inference. We unfold mean-field variational inference of **NPCRF** as a neural network for end-to-end training and inference. We find our method consistently outperforms its backbone, and reach competitive performance against very recent baselines on UFET and FET. **NPCRF** is efficient and require low additional computation costs. For future work, modeling higher-order label correlations, injecting prior knowledge into pairwise potentials, and extending **NPCRF** to other tasks are worth exploring.

Limitations

As shown in the experiments, the main limitation of **NPCRF** is that it has less positive effect on tasks that do not require understanding type correlation (e.g., tasks with small label sets and a low number of gold labels per instance), and on models that already model label semantics quite well (e.g., prompt-based methods). Another limitation of **NPCRF** is that, although it can be combined with many backbones, there still exist some backbones that cannot directly use **NPCRF**, such as models that generate types one by one in an autoregressive way (e.g., LRN) and models that cannot efficiently compute label logits (e.g., LITE).

7 Acknowledgement

This work was supported by the National Natural Science Foundation of China (61976139) and by Alibaba Group through Alibaba Innovative Research Program.

References

Anurag Arnab, Sadeep Jayasumana, Shuai Zheng, and Philip HS Torr. 2016. Higher order conditional random fields in deep neural networks. In *European conference on computer vision*, pages 524–540. Springer.

Siddhartha Chandra and Iasonas Kokkinos. 2016. Fast, exact and multi-scale inference for semantic image segmentation with deep gaussian crfs. In *European conference on computer vision*, pages 402–418. Springer.

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.

Eunsol Choi, Omer Levy, Yejin Choi, and Zettlemoyer. 2018. Ultra-fine entity typing. In *Proceedings of the ACL*. Association for Computational Linguistics.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, et al. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*.

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. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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

Ning Ding, Guangwei Xu, Yulin Chen, Xiaobin Wang, Xu Han, Pengjun Xie, Haitao Zheng, and Zhiyuan Liu. 2021b. Few-NERD: A few-shot named entity recognition dataset. 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 3198–3213, Online. Association for Computational Linguistics.

Kunihiko Fukushima and Sei Miyake. 1982. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets*, pages 267–285. Springer.

Nadia Ghamrawi and Andrew McCallum. 2005. Collective multi-label classification. In *Proceedings of*

- the 14th ACM International Conference on Information and Knowledge Management, CIKM '05*, page 195–200, New York, NY, USA. Association for Computing Machinery.
- Dan Gillick, Nevena Lazic, Kuzman Ganchev, Jesse Kirchner, and David Huynh. 2014. Context-dependent fine-grained entity type tagging. *arXiv preprint arXiv:1412.1820*.
- Eduard Hovy, Mitchell 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, New York City, USA. Association for Computational Linguistics.
- Zechuan Hu, Yong Jiang, Nguyen Bach, Tao Wang, Zhongqiang Huang, Fei Huang, and Kewei Tu. 2020. **An investigation of potential function designs for neural CRF**. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2600–2609, Online. Association for Computational Linguistics.
- Hailong Jin, Lei Hou, Juanzi Li, and Tiansi Dong. 2019. **Fine-grained entity typing via hierarchical multi graph convolutional networks**. 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)*, pages 4969–4978, Hong Kong, China. Association for Computational Linguistics.
- Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.
- Daphne Koller and Nir Friedman. 2009. *Probabilistic graphical models: principles and techniques*. MIT press.
- Đ Khuê Lê-Huu and Karteek Alahari. 2021a. Regularized frank-wolfe for dense crfs: Generalizing mean field and beyond. *Advances in Neural Information Processing Systems*, 34.
- Đ Khuê Lê-Huu and Karteek Alahari. 2021b. Regularized frank-wolfe for dense crfs: Generalizing mean field and beyond. *Advances in Neural Information Processing Systems*, 34.
- Bangzheng Li, Wenpeng Yin, and Muhao Chen. 2022. Ultra-fine entity typing with indirect supervision from natural language inference. *arXiv preprint arXiv:2202.06167*.
- 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. 2021. **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.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- 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.
- Ilya Loshchilov and Frank Hutter. 2018. Fixing weight decay regularization in adam.
- Aaron Mueller, Jason Krone, Salvatore Romeo, Saab Mansour, Elman Mansimov, Yi Zhang, and Dan Roth. 2022. Label semantic aware pre-training for few-shot text classification. *arXiv preprint arXiv:2204.07128*.
- 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. 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, Volume 1 (Long and Short Papers)*, pages 2407–2417, Minneapolis, Minnesota. Association for Computational Linguistics.
- Weiran Pan, Wei Wei, and Feida Zhu. 2022. Automatic noisy label correction for fine-grained entity typing. *arXiv preprint arXiv:2205.03011*.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. **GloVe: Global vectors for word representation**. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.
- Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. **Deep contextualized word representations**. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

- Xiang Ren, Wenqi He, Meng Qu, Clare R Voss, Heng Ji, and Jiawei Han. 2016. 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*, pages 1825–1834.
- Falong Shen, Rui Gan, Shuicheng Yan, and Gang Zeng. 2017. Semantic segmentation via structured patch prediction, context crf and guidance crf. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1953–1961.
- 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.
- Martin J Wainwright, Michael I Jordan, et al. 2008. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305.
- Bingyu Wang, Cheng Li, Virgil Pavlu, and Javed Aslam. 2017. Regularizing model complexity and label structure for multi-label text classification. *arXiv preprint arXiv:1705.00740*.
- Ralph Weischedel and Ada Brunstein. 2005. Bbn pronoun coreference and entity type corpus. Technical Report LDC2005T33, linguistic Data Consortium.
- Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.
- 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. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 16–25, New Orleans, Louisiana. Association for Computational Linguistics.
- Li Yang and Yanhong Zhou. 2010. Two-phase biomedical named entity recognition based on semi-crfs. In *2010 IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA)*, pages 1061–1065.
- 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*, pages 149–159, Austin, Texas. Association for Computational Linguistics.
- Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. 2015. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE international conference on computer vision*, pages 1529–1537.

A Example Appendix

A.1 Pairwise Potentials

We recover the learned log potentials $\Theta_{00}, \Theta_{11}, \Theta_{01}, \Theta_{10}$ using Eq. 2. For visualization, we pick three sets of types, types in the same set are relevant, and types between different set are irrelevant, and draw the heatmap of log potentials of them in Fig. 3. The figure shows that: (1) the learned potentials obey the intrinsic properties, Θ_{00}, Θ_{11} are symmetric and different from each other, $\Theta_{01} = \Theta_{10}^T$ and they are asymmetric. (2) Θ_{00}, Θ_{11} do encode different levels of similarity, the scores representing the co-absence and co-occurrence of types in the same set are higher than types in different set. (3) The Θ_{01} and Θ_{10} obey the intrinsic properties and cells with high scores usually represent the co-exclusion of types from different sets. In general, Θ_{00} and Θ_{11} are less interpretable compared to Θ_{01} and Θ_{10} .

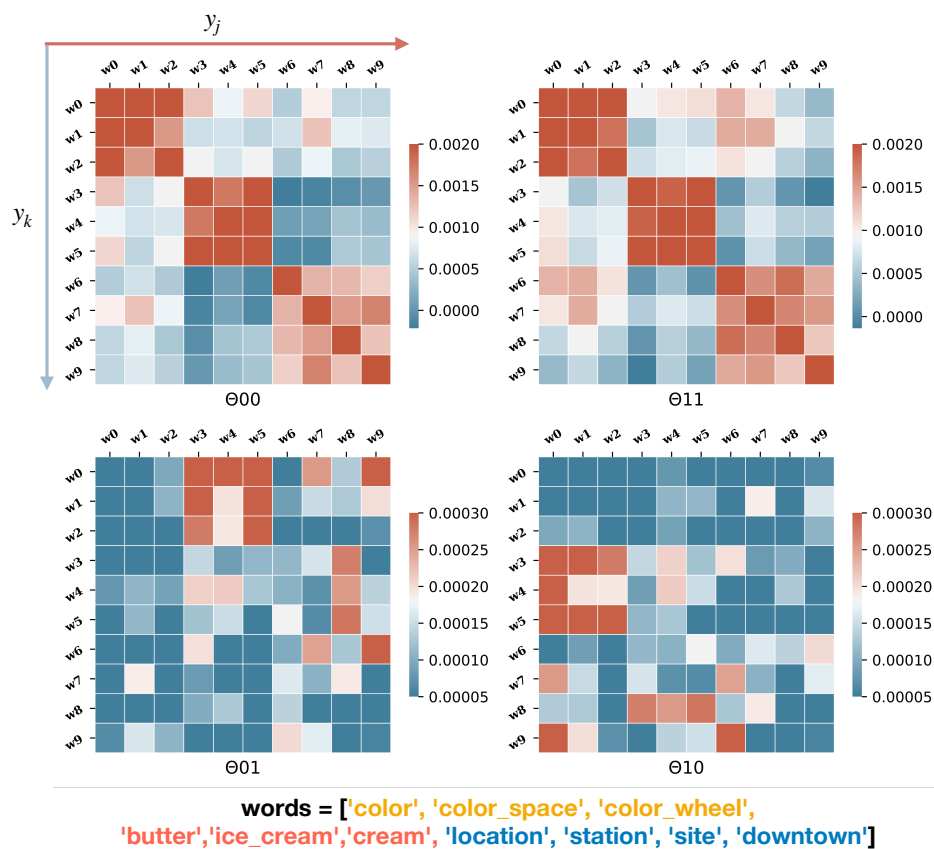


Figure 3: Visualization of pairwise potentials. The horizontal axis is for y_k , and the vertical axis is for y_j , for example, $\theta_p(\text{location} = 0, \text{downtown} = 1) = \Theta_{01}[6, 9]$.