Imposing Label-Relational Inductive Bias for Extremely Fine-Grained Entity Typing

Wenhan Xiong[†], **Jiawei Wu**[†], **Deren Lei**[†], **Mo Yu**[*], **Shiyu Chang**[*], **Xiaoxiao Guo**[*], **William Yang Wang**[†]
[†] University of California, Santa Barbara
[*] IBM Research
{xwhan, william}@cs.ucsb.edu, yum@us.ibm.com, {shiyu.chang, xiaoxiao.guo}@ibm.com

Abstract

Existing entity typing systems usually exploit the type hierarchy provided by knowledge base (KB) schema to model label correlations and thus improve the overall performance. Such techniques, however, are not directly applicable to more open and practical scenarios where the type set is not restricted by KB schema and includes a vast number of free-form types. To model the underlying label correlations without access to manually annotated label structures, we introduce a novel label-relational inductive bias, represented by a graph propagation layer that effectively encodes both global label co-occurrence statistics and word-level similarities. On a large dataset with over 10,000 free-form types, the graph-enhanced model equipped with an attention-based matching module is able to achieve a much higher recall score while maintaining a high-level precision. Specifically, it achieves a 15.3% relative F1 improvement and also less inconsistency in the outputs. We further show that a simple modification of our proposed graph layer can also improve the performance on a conventional and widely-tested dataset that only includes KB-schema types.[1]

1 Introduction

Fine-grained entity typing is the task of identifying specific semantic types of entity mentions in given contexts. In contrast to general entity types (*e.g.*, organization, event), fine-grained types (*e.g.*, political party, natural disaster) are often more informative and can provide valuable prior knowledge for a wide range of NLP tasks, such as coreference resolution (Durrett and Klein, 2014), relation extraction (Yaghoobzadeh et al., 2016) and question answering (Lee et al., 2006; Yavuz et al., 2016).

[1] https://github.com/xwhan/Extremely-Fine-Grained-Entity-Typing

Context	Types
Big Show then appeared at One Night Stand, attacking Tajiri, Super Crazy, and the Full Blooded Italians after their tag team match	person[†], television_program[⋆]
	person, athlete, wrestler, entertainer
The womens pole vault at the 2010 IAAF World Indoor Championships was held at the ASPIRE Dome on 12 and 14 **March**.	month[†], event[⋆]
	date, month

Table 1: Examples of inconsistent predictions produced by existing entity typing system that does not model label correlations. We use different subscript symbols to indicate contradictory type pairs and show the ground-truth types in *italics*.

In practical scenarios, a key challenge of entity typing is to correctly predict *multiple* ground-truth type labels from a large candidate set that covers a wide range of types in different granularities. In this sense, it is essential for models to effectively capture the inter-label correlations. For instance, if an entity is identified as a "criminal", then the entity must also be a "person", but it is less likely for this entity to be a "police officer" at the same time. When ignoring such correlations and considering each type separately, models are often inferior in performance and prone to inconsistent predictions. As shown in Table 1, an existing model that independently predicts different types fails to reject predictions that include apparent contradictions.

Existing entity typing research often address this aspect by explicitly utilizing a given type hierarchy to design hierarchy-aware loss functions (Ren et al., 2016b; Xu and Barbosa, 2018) or enhanced type label encodings (Shimaoka et al., 2017) that enable parameter sharing between related types. These methods rely on the assump-

Proceedings of NAACL-HLT 2019, pages 773–784
Minneapolis, Minnesota, June 2 - June 7, 2019.

tion that the underlying type structures are predefined in entity typing datasets. For benchmarks annotated with the knowledge base (KB) guided distant supervision, this assumption is often valid since all types are from KB ontologies and naturally follow tree-like structures. However, since knowledge bases are inherently incomplete (Min et al., 2013), existing KBs only include a limited set of entity types. Thus, models trained on these datasets fail to generalize to lots of unseen types. In this work, we investigate entity typing in a more open scenario where the type set is not restricted by KB schema and includes over 10,000 free-form types (Choi et al., 2018). As most of the types do not follow any predefined structures, methods that explicitly incorporate type hierarchies cannot be straightforwardly applied here.

To effectively capture the underlying label correlations without access to known type structures, we propose a novel label-relational inductive bias, represented by a graph propagation layer that operates in the latent label space. Specifically, this layer learns to incorporate a label affinity matrix derived from global type co-occurrence statistics and word-level type similarities. It can be seamlessly coupled with existing models and jointly updated with other model parameters. Empirically, on the Ultra-Fine dataset (Choi et al., 2018), the graph layer alone can provide a significant 11.9% relative F1 improvement over previous models. Additionally, we show that the results can be further improved ($11.9\% \rightarrow 15.3\%$) with an attention-based mention-context matching module that better handles *pronouns* entity mentions. With a simple modification, we demonstrate that the proposed graph layer is also beneficial to the widely used OntoNotes dataset, despite the fact that samples in OntoNotes have lower label multiplicity (*i.e.*, average number of ground-truth types for each sample) and thus require less label-dependency modeling than the Ultra-Fine dataset.

To summarize, our major contribution includes:

- We impose an effective label-relational bias on entity typing models with an easy-to-implement graph propagation layer, which allows the model to implicitly capture type dependencies;
- We augment our graph-enhanced model with an attention-based matching module, which constructs stronger interactions between the mention and context representations;
- Empirically, our model is able to offer significant improvements over previous models on the Ultra-Fine dataset and also reduces the cases of inconsistent type predictions.

2 Related Work

Fine-Grained Entity Typing The task of fine-grained entity typing was first thoroughly investigated in (Ling and Weld, 2012), which utilized Freebase-guided distant supervision (DS) (Mintz et al., 2009) for entity typing and created one of the early large-scale datasets. Although DS provides an efficient way to annotate training data, later work (Gillick et al., 2014) pointed out that entity type labels induced by DS ignore entities' local context and may have limited usage in context-aware applications. Most of the following research has since focused on testing in context-dependent scenarios. While early methods (Gillick et al., 2014; Yogatama et al., 2015) on this task rely on well-designed loss functions and a suite of handcraft features that represent both context and entities, Shimaoka et al. (2016) proposed the first attentive neural model which outperformed feature-based methods with a simple cross-entropy loss.

Modeling Entity Type Correlations To better capture the underlying label correlations, Shimaoka et al. (2017) employed a hierarchical label encoding method and AFET (Ren et al., 2016a) used the predefined label hierarchy to identify noisy annotations and proposed a partial-label loss to reduce such noise. A recent work (Xu and Barbosa, 2018) proposed hierarchical loss normalization which alleviated the noise of too specific types. Our work differs from these works in that we do not rely on known label structures and aim to learn the underlying correlations from data. Rabinovich and Klein (2017) recently proposed a structure-prediction approach which used type correlation features. The inference on their learned factor graph is approximated by a greedy decoding algorithm, which outperformed unstructured methods on their own dataset. Instead of using an explicit graphical model, we enforce a relational bias on model parameters, which does not introduce extra burden on label decoding.

3 Task Definition

Specifically, the task we consider takes a raw sentence C as well as an entity mention span M inside

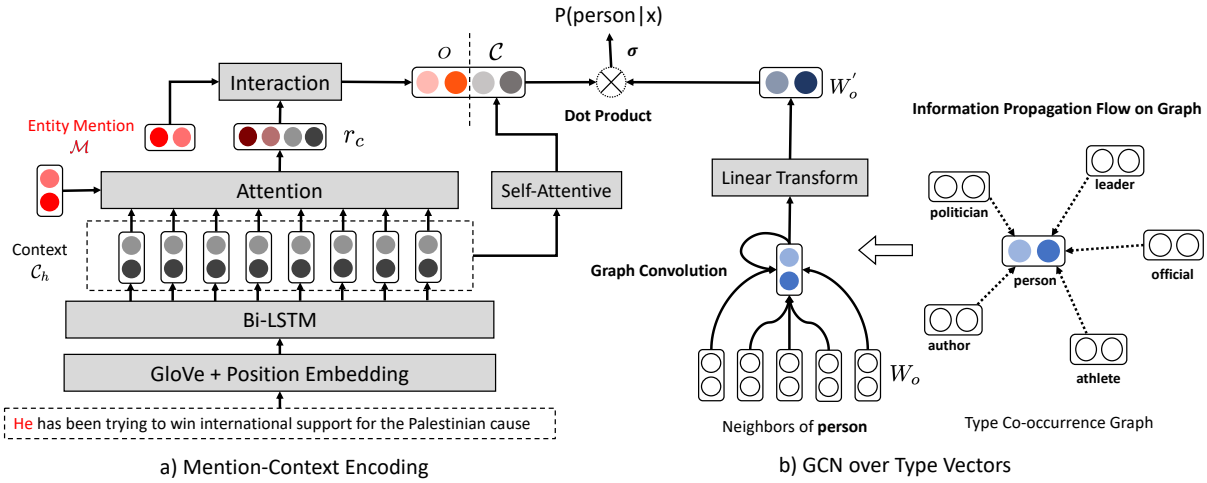


Figure 1: Overview of the process to make predictions on the type **"person"**. **a)** Modules used to extract mention and context aware representations. **b)** An illustration of the graph layer operating over the type vector of **"person"**.

C as inputs, and aims to predict the correct type labels T_m of M from a candidate type set $\mathcal{T}$, which includes more than 10,000 free-form types. The entity span M here can be named entities, nominals and also pronouns. The ground-truth type set T_m here usually includes more than one types (approximately five types on average), making this task a multi-label classification problem.

4 Methodology

In this section, we first briefly introduce the neural architecture to encode raw text inputs. Then we describe the matching module we use to enhance the interaction between the mention span and the context sentence. Finally, we move to the label decoder, on which we impose the label-relational bias with a graph propagation layer that encodes type co-occurrence statistics and word-level similarities. Figure 1 provides a graphical overview of our model, with 1a) illustrating both the text encoders and the matching module, and 1b) showing an example of graph propagation.

4.1 Representation Model

Our base model to encode the context and the mention span follows existing neural approaches (Shimaoka et al., 2016; Xu and Barbosa, 2018; Choi et al., 2018). To encode the context, we first apply a standard Bi-LSTM, which takes GloVe (Pennington et al., 2014) embeddings and position embeddings (three vectors representing positions before, inside or after the mention span) as inputs and outputs the hidden states at each time step $t \in [1, l_c]$. With the derived hidden states $\mathcal{C}_h \in \mathbb{R}^{l_c \times h_c}$, we then apply a self-attentive encoder (McCann et al., 2017) on the top to get the final context representation $\mathcal{C}$. For the entity mention span, we concatenate the features derived by a character-level CNN and a similar self-attentive encoder. We denote the final mention representation as $\mathcal{M}$.[2]

4.2 Mention-Context Interaction

Since most previous datasets only consider *named entities*, a simple concatenation of the two features $[\mathcal{C}; \mathcal{M}]$ followed by a linear output layer (Shimaoka et al., 2016, 2017) usually works reasonably well when making predictions. This suggests that $\mathcal{M}$ itself provides important information for recognizing entity types. However, as in our target dataset, a large portion of entity mentions are actually *pronouns*, such as "he" or "it", this kind of mentions alone provide only limited clues about general entity types (*e.g.*, "he" is a "person") but little information about fine-grained types. In this case, directly appending representation of pronouns does not provide extra useful information for making fine-grained predictions. Thus, instead of using the concatenation operator, we propose to construct a stronger interaction between the mention and context with an attention-based matching module, which has shown its effectiveness in recent natural language inference models (Mou et al., 2016; Chen et al., 2017).

Formally consider the mention representation $\mathcal{M} \in \mathbb{R}^{h_m}$ and context's hidden feature $\mathcal{C}_h \in$

[2]Please refer to (Shimaoka et al., 2017) and (Choi et al., 2018) for more detailed descriptions.

$\mathbb{R}^{l_c \times h_c}$, where l_c indicates the number of tokens in the context sentence and h_m, h_c denote feature dimensions. We first project the mention feature $\mathcal{M}$ into the same dimension space as $\mathcal{C}_h$ with a linear layer ($W_1 \in \mathbb{R}^{h_m \times h_c}$) and a $tanh$ function[3]:

$$m_{proj} = tanh(W_1^T \mathcal{M}), \quad (1)$$

then we perform bilinear attention matching between m_{proj} and $\mathcal{C}_h$, resulting in an affinity matrix $\mathcal{A}$ with dimension $\mathcal{A} \in \mathbb{R}^{1 \times l_c}$:

$$\mathcal{A} = m_{proj} \times W_a \times \mathcal{C}_h, \quad (2)$$

where $W_a \in \mathbb{R}^{h_c \times h_c}$ is a learnable matrix. If we consider the mention feature as query and the context as memory, we can use the affinity matrix to retrieve the relevant parts in the context:

$$\bar{\mathcal{A}} = softmax(\mathcal{A}) \quad (3)$$

$$r_c = \bar{\mathcal{A}} \times \mathcal{C}_h. \quad (4)$$

With the projected mention representation m_{proj} and the retrieved context feature r_c, we define the following interaction operators:

$$r = \rho(W_r[r_c; m_{proj}; r_c - m_{proj}]) \quad (5)$$

$$g = \sigma(W_g[r_c; m_{proj}; r_c - m_{proj}]) \quad (6)$$

$$o = g * r + (1 - g) * m_{proj}, \quad (7)$$

where $\rho(\cdot)$ is a gaussian error linear unit (Hendrycks and Gimpel, 2016) and r is the fused context-mention feature; $\sigma(\cdot)$ indicates a *sigmoid* function and g is the resulting gating function, which controls how much information in mention span itself should be passed down. We expect the model to focus less on the mention representation when it is not informative. The concatenation $[r_c; m_{proj}; r_c - m_{proj}]$ here is supposed to capture different aspects of the interactions. To emphasize the context's impact, we finally concatenate the extracted context feature ($\mathcal{C}$) with the output (o) of the matching module ($f = [o; \mathcal{C}]$) for prediction.

4.3 Imposing Label-Relational Inductive Bias

For approaches that ignore the underlying label correlations, the type predictions are considered as N independent binary classification problems, with N being the number of types. If we denote the feature extracted by any arbitrary neural model as $f \in \mathbb{R}^{d_f}$, then the probability of being any given type is calculated by:

$$p = \sigma(W_o f), W_o \in \mathbb{R}^{N \times d_f}. \quad (8)$$

We can see that every row vector of W_o is responsible for predicting the probability of one particular type. We will refer the row vectors as type vectors for the rest of this paper. As these type vectors are independent, the label correlations are only implicitly captured by sharing the model parameters that are used to extract f. We argue that the paradigm of parameter sharing is not enough to impose strong label dependencies and the values of type vectors should be better constrained.

A straightforward way to impose the desired constraints is to add extra regularization terms on W_o. We first tested several auxiliary loss functions based on the heuristics from GloVe (Pennington et al., 2014), which operates on the type co-occurrence matrix. However, the auxiliary losses only offer trivial improvements in our experiments. Instead, we find that directly imposing a model-level inductive bias on the type vectors turns out to be a more principled solution. This is done by adding a graph propagation layer over randomly initialized W_o and generating the updated type vectors W_o', which is used for final prediction. Both W_o and the graph convolution layer are learned together with other model parameters. We view this layer as the key component of our model and use the rest of this section to describe how we create the label graph and compute the propagation over the graph edges.

Label Graph Construction In KB-supervised datasets, the entity types are usually arranged in tree-like structures. Without any prior about type structures, we consider a more general graph-like structure. While the nodes in the graph straightforwardly represent entity types, the meaning of the edges is relatively vague, and the connections are also unknown. In order to create meaningful edges using training data as the only resource, we utilize the type co-occurrence matrix: if two type t_1 and t_2 both appear to be the true types of a particular entity mention, we will add an edge between them. In other words, we are using the co-occurrence statistics to approximate the pair-wise dependencies and the co-occurrence matrix now serves as the adjacent matrix. Intuitively, if t_2 co-appears with t_1 more often than another type t_3, the probabilities of t_1 and t_2 should have stronger depen-

[3] $tanh$ here is used to make m_{proj} in the same scale as $\mathcal{C}_h$, which was the output of a $tanh$ function inside LSTM.

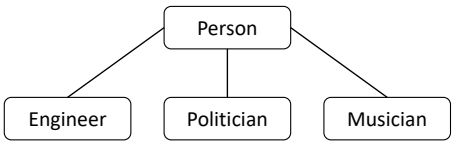


Figure 2: A snippet of the underlying type co-occurrence graph. Multiple edges between nodes are omitted here for clarity.

dencies and the corresponding type vectors should be more similar in the vector space. In this sense, we expect each type vector to effectively capture the local neighbor structure on the graph.

Correlation Encoding via Graph Convolution To encode the neighbor information into each node's representation, we follow the propagation rule defined in Graph Convolution Network (GCN) (Kipf and Welling, 2016). In particular, with the adjacent or co-occurrence matrix A, we define the following propagation rule on W_o:

$$W_o^{'} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}W_oT \quad (9)$$

$$\tilde{A} = A + I_N. \quad (10)$$

Here $T \in \mathbb{R}^{d_f \times d_f}$ is the transformation matrix and I_N is an identity matrix used to add self-connected edges. $\tilde{D}$ is a diagonal degree matrix with $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$, which is used to normalize the feature vectors such that the number of neighbors does not affect the scale of transformed feature vectors. In our experiments, we find that an alternative propagation rule

$$W_o^{'} = \tilde{D}^{-1}\tilde{A}W_oT \quad (11)$$

works similarly well and is more efficient as it involves less matrix multiplications. If we look closely and take each node out, the propagation can be written as

$$W_o^{'}[i,:] = \frac{1}{\sum_j \tilde{A}_{ij}}(\sum_j \tilde{A}_{ij}W_o[j,:]T). \quad (12)$$

From this formula, we can see that the propagation is essentially gathering features from the first-order neighbors. In this way, the prediction on type t_i is dependent on its neighbor types.

Compared to original GCNs that often use multi-hop propagations (*i.e.*, multiple graph layers connected by nonlinear functions) to capture higher-order neighbor structures. We only apply one-hop propagation and argue that high-order label dependency is not necessarily beneficial in our scenario and might introduce false bias. A simple illustration is shown in Figure 2. We can see that propagating 2-hop information introduces undesired inductive bias, since types that are more than 1-hop away (*e.g.*, "Engineer" and "Politician") usually do not have any dependencies. In fact, some of the 2-hop type pairs can be contradictory types (*e.g.*, "police" and "prisoner"). This hypothesis is consistent with our experiment results: adding more than one graph layer leads to worse results. Additionally, we also omit GCN's nonlinear activation which introduces unnecessary constraints on the scale of $W_o^{'}$, with which we calculate the unscaled scores before calculating the probability via a sigmoid function.

4.4 Leveraging Label Word Embeddings

As the type labels are all written as text phrases, an interesting question is whether we can exploit the semantics provided by pre-trained word embeddings to improve entity typing. We explore this possibility by using the cosine similarity of word embeddings. We first calculate type embeddings by simply summing the embeddings of all tokens in the type name. Then we build a label affinity matrix A_{word} by calculating pair-wise cosine similarities. With the assumption that word-level similarity measures some degree of label dependency, we propose to integrate A_{word} into the graph convolution layer following

$$A_{word}^{'} = (A_{word} + 1)/2 \quad (13)$$

$$W_o^{'} = \tilde{D}^{-1}(\tilde{A} + \lambda A_{word}^{'})W_oT. \quad (14)$$

Here Equation 13 scales the similarity value into $(0, 1]$ to avoid negative edge weights, which might introduce numerical issues when calculating $\tilde{D}^{-1}$. λ is a trainable parameter used to weight the impact of word-level similarities. As will be shown in Section 5, this simple augmentation provides further improvement over our original model.

5 Experiments

5.1 Experiment Setup

Datasets Our experiments mainly focus on the Ultra-Fine entity typing dataset which has 10,331 labels and most of them are defined as free-form text phrases. The training set is annotated with heterogeneous supervisions based on KB, Wikipedia and head words in dependency trees,

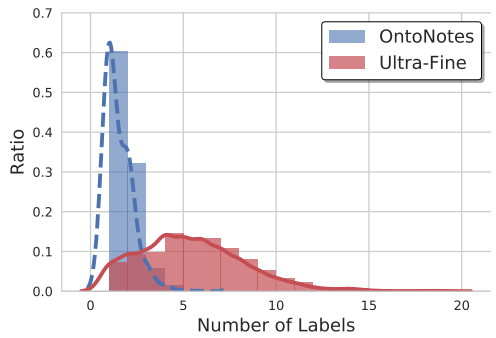


Figure 3: Label multiplicity distribution of the datasets.

resulting in about 25.2M[4] training samples. This dataset also includes around 6,000 crowdsourced samples. Each of these samples has five ground-truth labels on average. For a fair comparison, we use the original test split of the crowdsourced data for evaluation. To better understand the capability of our model, we also test our model on the commonly-used OntoNotes (Gillick et al., 2014) benchmark. It is worth noting that this dataset is much smaller and has lower label multiplicity than the Ultra-Fine dataset, *i.e.*, each sample only has around 1.5 labels on average. Figure 3 shows a comparison of these two datasets.

Baselines For the Ultra-Fine dataset, we compare our model with AttentiveNER (Shimaoka et al., 2016) and the multi-task model proposed with the Ultra-Fine dataset. Note that *other models that require pre-defined type hierarchy are not applicable to this dataset*. For experiments on OntoNotes, in addition to the two neural baselines for Ultra-Fine, we compare with several existing methods that explicitly utilize the pre-defined type structures in loss functions. Namely, these methods are AFET (Ren et al., 2016a), LNR (Ren et al., 2016b) and NFETC (Xu and Barbosa, 2018).

Evaluation Metrics On Ultra-Fine, we first evaluate the mean reciprocal rank (MRR), macro precision(P), recall (R) and F1 following existing research. As P, R and F1 all depend on a chosen threshold on probabilities, we also consider a more transparent comparison using precision-recall curves. On OntoNotes, we use the standard metrics used by baseline models: accuracy, macro, and micro F1 scores.

[4]Choi et al. (2018) use the licensed Gigaword to build part of the dataset, while in our experiments we only use the open-sourced training set which has approximately 6M training samples.

Implementation Details Most of the model hyperparameters, such as embedding dimensions, learning rate, batch size, dropout ratios on context and mention representations are consistent with existing models. Since the mention-context matching module brings more parameters, we apply a dropout layer over the extracted feature f to avoid overfitting. We list all the hyperparameters in the appendix. Models for OntoNotes are trained with standard binary cross-entropy (BCE) losses defined on all candidate labels. When training on Ultra-Fine, we adopt the multi-task loss proposed in Choi et al. (2018) which divides the cross-entropy loss into three separate losses over different type granularities. The multi-task objective avoids penalizing false negative types and can achieve higher recalls.

5.2 Evaluation on the Ultra-Fine Dataset

We report the results on Ultra-Fine in Table 2. It is worth mentioning that our model, denoted as LABELGCN, is trained using the unlicensed training set which is smaller than the one used by compared baselines. Even though our model significantly outperforms the baselines, for a fair comparison, we first test our model using the same decision threshold (0.5) used by previous models. In terms of F1, our best model (LABELGCN) outperforms existing methods by a large margin. Compared to Choi et al. (2018), our model improves on both precision and recall significantly. Compared to the AttentiveNER trained with standard BCE loss, our model achieves much higher recall but performs worse in precision. This is due to the fact that when trained with BCE loss, the model usually retrieves only one label per sample and these types are mostly general types[5] which are easier to predict. With higher recalls or more retrieved types, achieving high precision requires being accurate on fine-grained types, which are often harder to predict.

As the precision and recall scores both rely on the decision threshold, different models or different metrics can have different optimal thresholds. As shown by the "LABELGCN + thresh tuning" entry in Table 2, with threshold tuning, our model beats baselines in all metrics. We also see that recall is usually lagging behind precision on this dataset, indicating that F1 score is mainly affected

[5]According to the results of our own implementation of BCE-trained model which achieves similar performance as AttentiveNER.

Model	Dev				Test			
	MRR	P	R	F1	MRR	P	R	F1
AttentiveNER	0.221	53.7	15.0	23.5	0.223	54.2	15.2	23.7
Choi et al. (2018)	0.229	48.1	23.2	31.3	0.234	47.1	24.2	32.0
LABELGCN	**0.250**	50.5	**28.7**	**36.6**	**0.253**	50.3	**29.2**	**36.9**
- w/o word embedding	0.245	49.4	27.8	35.6	0.249	48.7	28.3	35.8
- w/o gcn propagation	0.231	47.8	25.7	33.5	0.239	45.4	25.8	32.9
- w/o mention-context interaction	0.249	53.2	25.0	34.0	0.253	54.3	25.8	35.0
LABELGCN + *threshold-tuning*	**0.250**	**55.6**	25.4	35.0	**0.253**	**54.8**	25.9	35.1

Table 2: Comparison with baseline models on the Ultra-Fine dataset. *Threshold-tuning* gives better performance on all metrics compared to both baselines.

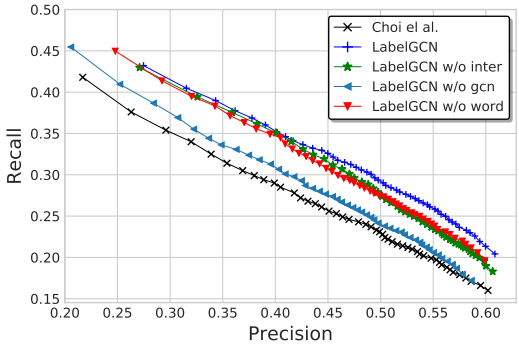


Figure 4: Precision-recall curves on Ultra-Fine. The trivial point derived by threshold 0 is omitted here.

Model	F1-pronouns	F1-else
Choi et al. (2018)	35.8	32.0
Choi et al. (2018) + inter	38.2 (↑ 2.4)	32.8
LABELGCN w/o inter	38.6	36.8
LABELGCN	39.3 (↑ 0.7)	36.5

Table 3: Decomposed validation performance on pronouns and the other entities. Each entry is obtained using the best threshold among the 50 equal-interval thresholds. The corresponding PR curves can be found in the appendix (Figure 5).

by the recall and tuning towards recall can usually lead to higher F1 scores. For more transparent comparisons, we show the precision-recall curves in Figure 4. These data points are based on the validation performance given by 50 equal-interval thresholds between 0 and 1. We can see there is a clear margin between our model and the multi-task baseline method (LabelGCN vs Choi et al.).

5.3 Ablation Studies

To quantify the effect of different model components, we report the performance of model variants in Table 2 and Figure 4. We can clearly see that the graph convolution layer is the most essential component. The information provided by word embedding is useful and can further improve both precision and recall. Although Table 2 seems to indicate the interaction module decreases the precision, we can see from Figure 4 that with a proper threshold, the enhanced interaction actually improves both precision and recall. In term of this, we recommend future research to use PR curves for more accurate model analysis.

5.4 Fine-Grained Performance for Pronouns

As discussed in Section 4.2, the mention representation of pronouns provide limited information about fine-grained types. We investigate the effect of the enhanced mention-context interaction by analyzing the decomposed performance on pronouns and other kinds of entities. From the results in Table 3, we can see that the enhanced interaction offers consistent improvements over pronouns entities and also maintains the performance on other kinds of entities.

5.5 Qualitative Analysis

To gain insights on the improvements provided by our model, we manually analyze 100 error cases[6] of the baseline model (Choi et al. (2018) with threshold 0.5) and see if our model can generate high-quality predictions. We first observe that many errors actually results from incomplete annotations. This suggests models' precision scores are often underestimated in this dataset. We discuss several typical error cases shown in Table 4 and list more samples in the appendix (Table 7).

A key observation is that while the baseline model tends to make inconsistent predictions (see examples 1, 2, 3), our model can avoid predicting such inconsistent type pairs. This indeed validates our model's ability to encode label correlations. We also notice that our model is more sensitive to gender information indicated by pronouns, while

[6]The baseline model achieves the lowest precision on these 100 samples.

1) Context	**Today**, Taiwan is manifesting the elegance of a democratic island, once again attracting global attention, as the people on this land create a new page in our history.
Groundtruth	*time, date, day, today, present*
Prediction	**Baseline**: {day†, person⋆, organization, religion} **Ours**: {day}
2) Context	**A gigantic robot** emerges, emitting a sound that paralyzes humans and disrupts all electrical systems in New York City.
Groundtruth	*object, device, machine, mechanism*
Prediction	**Baseline**: {object, person†, robot⋆} **Ours**: {object, robot}
3) Context	**He** also has been accused of genocide in Bosnia and other war crimes in Croatia, but the date to try those two indictments together has not been set.
Groundtruth	*person*
Prediction	**Baseline**:{person, god†, title, criminal⋆} **Ours**: {person, politician, criminal, male, prisoner}
4) Context	**Her** status was uncertain for Wimbledon, which begins June 23.
Groundtruth	*person, athlete, adult, player, professional, tennis player, contestant*
Prediction	**Baseline**: {person, female, woman, spouse} **Ours**: {person, artist, female, woman}
5) Context	For eight years **he** treated thousands of wounded soldiers of the armed forces led by the CPC.
Groundtruth	*person, doctor, caretaker, nurse*
Prediction	**Baseline**: {person, soldier, suspect, serviceman} **Ours**: {person, soldier, man}

Table 4: Qualitative analysis of validation samples. We use different colors and subscript symbols to mark inconsistencies. The bottom two rows show error cases for both models.

Model	Accuracy	Macro-F1	Micro-F1
AttentiveNER	51.7	71.0	64.9
AFET	55.1	71.1	64.7
LNR	57.2	71.5	66.1
NFETC	**60.2**	76.4	70.2
Choi et al. (2018)	59.5	76.8	71.8
LABELGCN	59.6	**77.8**	**72.2**

Table 5: Results on OntoNotes. Upper rows show the results of baselines that explicitly use the hierarchical type structures.

the baseline model sometimes holds the gender-indicating predictions and predict other types, our model predicts the gender-indicating types more often (examples 3, 4, 5). We conjecture that our model learns this easy way to maintain precision.

For cases that both models fail, some of them actually require background knowledge (example 4) to make accurate predictions. Another typical case is that both models predict some other entities in the context (example 5). We think this potentially results from the data bias introduced by the head-word supervision.

5.6 Evaluation on OntoNotes

To better understand the requirements for applying our model, we further evaluate on the OntoNotes dataset. Here we do not apply the proposed mention-context matching module as this dataset does not include any pronoun entities. To obtain more reliable co-occurrence statistics, we use the augmented training data released by Choi et al. (2018). However, since the training set is still much smaller than that of the Ultra-Fine dataset, the derived co-occurrence statistics are relatively noisy and might introduce undesired bias. We thus add an additional residual connection to our graph convolution layer, which allows the model to selectively use co-occurrence statistics. This indeed gives us improvements over previous state-of-the-arts, as shown in Table 5. However, compared to Ultra-Fine, the margin of the improvement is smaller. In view of the key differences of these two datasets, we highlight two key requirements for our proposed model to offer substantial improvements. First, there should be a large-scale training set so that the derived co-occurrence statistics can reasonably reflect the true label correlations. Second, the samples themselves should also have higher label multiplicity. In fact, most of the samples in OntoNotes only have 1 or 2 labels. This property actually alleviates the need for models to capture label dependencies.

6 Conclusion

In this paper, we present an effective method to impose label-relational inductive bias on fine-grained entity typing models. Specifically, we utilize a graph convolution layer to incorporate type co-occurrence statistics and word-level type similarities. This layer implicitly captures the label correlations in the latent vector space. Along with an attention-based mention-context matching module, we achieve significant improvements over

previous methods on a large-scale dataset. As our method does not require external knowledge about the label structures, we believe our method is general enough and has the potential to be applied to other multi-label tasks with plain-text labels.

Acknowledgement

This research was supported in part by DARPA Grant D18AP00044 funded under the DARPA YFA program. The authors are solely responsible for the contents of the paper, and the opinions expressed in this publication do not reflect those of the funding agencies.

References

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced lstm for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1657–1668. Association for Computational Linguistics.

Eunsol Choi, Omer Levy, Yejin Choi, and Luke Zettlemoyer. 2018. Ultra-fine entity typing. *arXiv preprint arXiv:1807.04905*.

Greg Durrett and Dan Klein. 2014. A joint model for entity analysis: Coreference, typing, and linking. *Transactions of the Association for Computational Linguistics*, 2:477–490.

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

Dan Hendrycks and Kevin Gimpel. 2016. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *arXiv preprint arXiv:1606.08415*.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

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

Xiao Ling and Daniel S Weld. 2012. Fine-grained entity recognition. In *AAAI*, volume 12, pages 94–100.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, pages 6294–6305.

Bonan Min, Ralph Grishman, Li Wan, Chang Wang, and David Gondek. 2013. Distant supervision for relation extraction with an incomplete knowledge base. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 777–782.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics.

Lili Mou, Rui Men, Ge Li, Yan Xu, Lu Zhang, Rui Yan, and Zhi Jin. 2016. Natural language inference by tree-based convolution and heuristic matching. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 130–136. Association for Computational Linguistics.

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.

Maxim Rabinovich and Dan Klein. 2017. Fine-grained entity typing with high-multiplicity assignments. *arXiv preprint arXiv:1704.07751*.

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*, pages 1369–1378.

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*, pages 1825–1834. ACM.

Sonse Shimaoka, Pontus Stenetorp, Kentaro Inui, and Sebastian Riedel. 2016. An attentive neural architecture for fine-grained entity type classification. In *Proceedings of the 5th Workshop on Automated Knowledge Base Construction*, pages 69–74. Association for Computational Linguistics.

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

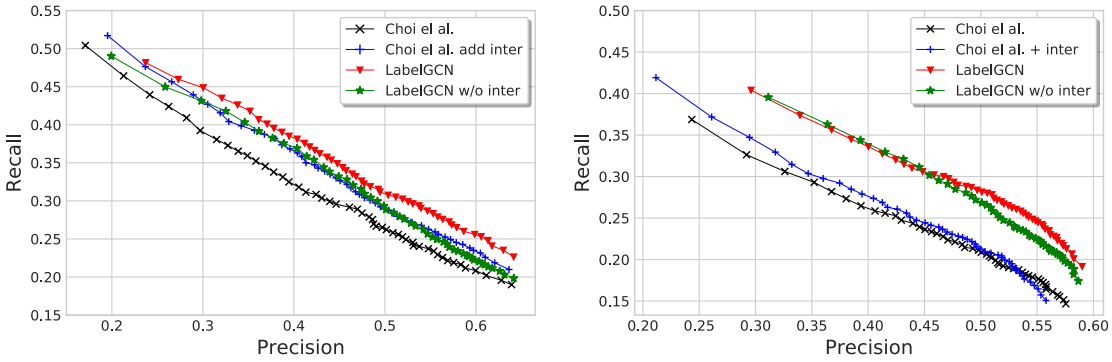
Peng Xu and Denilson Barbosa. 2018. Neural fine-grained entity type classification with hierarchy-aware loss. *arXiv preprint arXiv:1803.03378*.

Yadollah Yaghoobzadeh, Heike Adel, and Hinrich Schütze. 2016. Noise mitigation for neural entity typing and relation extraction. *arXiv preprint arXiv:1612.07495*.

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.

Dani Yogatama, Daniel Gillick, and Nevena Lazic. 2015. Embedding methods for fine grained entity type classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, volume 2, pages 291–296.

A Appendix



a) PR curves of pronoun entities

a) PR curves of non-pronoun entities

Figure 5: Precision-recall curves showing the decomposed results on pronoun and non-pronoun entity mentions. The enhanced mention-context interaction can consistently offer improvements for pronoun entity mentions while maintaining the performance for non-pronoun entity mentions.

learning rate	0.001
batch size	1000
position embedding size	50
dropout on context $\mathcal{C}$	0.2
dropout on mention $\mathcal{M}$	0.5
hidden dimension of LSTM	100
dropout on fused feature f (Ultra-Fine)	0.2
dropout on fused feature f (OntoNotes)	0.3

Table 6: Hyperparameters used in our experiments

Context	**They** have been asked to appear in court to face the charge on Feb. 3.
Groundtruth	*person, defendant, suspect, accused*
Prediction	**Baseline**:{person, engineer, officer, policeman†, prisoner⋆, married, serviceman} **Ours**:{person}
Context	"**It** is truly a war crime," she added.
Groundtruth	*event, crime, issue, offense, transgression, atrocity*
Prediction	**Baseline**: {internet†, event, art⋆, writing} **Ours**: {law}
Context	**She** added that Israeli military personnel had conducted a medical examination after the shooting in concert with Palestinian medics.
Groundtruth	*person*
Prediction	**Baseline**: {person†, art⋆, writing, convict, felon} **Ours**: {person, female, woman}
Context	The monument is located in Pioneer Park Cemetery in the Convention Center District of downtown Dallas, Texas, **USA**, next to the Dallas Convention Center and Pioneer Plaza.
Groundtruth	*location, place, country, area, nation, region*
Prediction	**Baseline**: {location†, person⋆, agency, artist, cemetery, country, language, title, republic} **Ours**:{nationality, location, place, country, area, license, nation}
Context	**The committee** undertook its work on Saturday 16/2/1426 A. H . The following is noteworthy :
Groundtruth	*group, organization, agency, company, institution, administration, body, management, party*
Prediction	**Baseline**: {committee†, person⋆, organization, government} **Ours**: {group, government, committee}
Context	**They** are accused of helping Libya develop a nuclear weapons programme and were alleged to have been in contact with Abdul Qadeer Khan , the disgraced father of Pakistan 's nuclear programme.
Groundtruth	*group, terrorist*
Prediction	**Baseline**: {military, person†, group, country⋆} **Ours**: {person, politician, prisoner, serviceman}
Context	**It** also marked the first major roundup of Islamist leaders by a government eager to demonstrate its commitment to the anti-terror fight waged by the United States.
Groundtruth	*event, consequence*
Prediction	**Baseline**:{internet†, event, art⋆, writing} **Ours**:{event}
Context	If you have ever watched a keynote speech by **Steve Jobs**, you know that he was the best of the best in launching a product.
Groundtruth	*person, adult, businessman, celebrity, professional*
Prediction	**Baseline**: {person, artist, athlete†, author, musician⋆} **Ours**:{person}
Context	**They** dined together, this time in Benedict's house, before the pope was driven back to his temporary residence in Regensburg 's St Wolfgang Seminary.
Groundtruth	*adult, man, supporter, serviceman*
Prediction	**Baseline**: {person, adult, female, woman} **Ours**: {person}
Context	Topic : **I** am grateful to the University of Science and Technology
Groundtruth	*person, individual, student*
Prediction	**Baseline**:{person, politician, employee, leader, minister, traveler, announcer, clergyman} **Ours**:{person, student}
Context	"**I** didn't think the speech was that long," Pataki said.
Groundtruth	*person, speaker*
Prediction	**Baseline**: {person, actor, politician, spokesperson, woman} **Ours**: {person, adult}
Context	"This is touching our troops," **she** said.
Groundtruth	*person, adult, female, reporter, woman*
Prediction	**Baseline**: {person, politician, official, spokesperson, communicator} **Ours**: {female, official, reporter, strategist, communicator, officeholder}

Table 7: More sample predictions. Our model is able to give more accurate type predictions and also reduce the inconsistency in the output type set.