ISSN 1330-3651(Print), ISSN 1848-6339 (Online)

https://doi.org/10.17559/TV-20250224002415

Original scientific paper

Entity Type-Aware Ultra-Fine Entity Typing with Adaptive Distance Optimization

Feng WANG*, YiXiu QIN

Abstract: Recent work have focused on representing entity mentions as well as entity types in a high-dimensional box space to more effectively capture their complex relationships. In box space, entity mentions as well as entity types are represented as high-dimensional hyper-rectangles. However, the role of entity types is often overlooked or inadequately incorporated in classification tasks within box space. Furthermore, the model struggles with precise optimization under specific conditions, or its constraints are excessively strict, leading to the complete overlap of the two box center points. In light of these shortcomings, this paper presents the Entity Type-aware Ultra-fine Entity Typing with Distance Optimization. The ETUT effectively integrates entity type information and introduces an adaptive distance-based module, ensuring model optimization even when two boxes are either fully separated or completely overlapped in space. Experimental results from ultra-fine and fine-grained entity typing datasets demonstrate the effectiveness of ETUT, showing it to be a state-of-the-art method in the domain of ultra-fine and fine-grained entity typing.

Keywords: fine-grained entity typing; multi-objective optimization; type hierarchy; ultra-fine entity typing

1 INTRODUCTION

Detecting entity types associated with entity located in the sentence is essential for text information analysis [1, 2], significantly impacting downstream NLP tasks [3] like knowledge base completion [4], co-reference resolution [5], entity linking [6, 7], relation extraction [8-10] and question answering [11]. Named entity recognition (NER) [12, 13] traditionally identify a small set of entity types (typically fewer than ten), which are primarily coarse-grained [12], such as entity type /person, entity type /location, as well as entity type /organization [2]. However, these entity types provided by NER are insufficient to deliver the fine-grained information required for downstream tasks [2, 14, 15].

The objective of ultra-fine (UFET) and fine-grained entity typing (FGET) is aims to associate an entity mention with multiple entity types based on both the entity mention and its context (UFET is a sub-task of FGET), overcoming the constraint of NER, which identifies only coarse-grained entity types. FGET enables an entity mention to be associated with more than one entity types, encompassing coarse-grained as well as fine-grained entity types [16], such as entity type /person, entity type /person/artist, and entity type /person/actor, making it more challenging than traditional NER [15, 17]. NER first integrates knowledge base relational information with textual attention mechanisms, overcoming the limitations of traditional models in modeling entity-context correlations.

Since FGET involves fine-grained entity types, it necessitates a large number of entity types to maintain classification precision [15, 16, 18]. Due to the inherent complexity of FGET, a large number of entity types are necessary [15, 16, 19], typically ranging from hundreds [14] to thousands. To organize these entity types, entity types in FGET often follow intricate structures, like type hierarchy [17, 20-22]. Researchers have shown considerable interest in the type hierarchy of FGET [14, 17, 20, 22-26]. Numerous studies have demonstrated that the type hierarchy significantly enhances the performance of FGET [17, 22-24]. Previous studies have incorporated type hierarchy via techniques such as hierarchy loss [23], complex bilinear mapping [24], and hyperbolic space [25]. It is crucial to highlight that techniques such as hierarchy loss [23], learning-to-rank loss [22], and complex bilinear mapping [24] depend on an explicit type hierarchy, which is absent in some FGET datasets.

UFET methods utilizing hyperbolic space [25, 28, 29] do not rely on an explicit type hierarchy, but they suffer from a lack of mathematical interpretation and are limited in their ability to capture entity type relations [18]. In order to overcome this shortcomings, Onoe et al. [18] introduced the novel approach of modeling UFET in high-dimensional box space. This method maps the representations of entity mentions and types as well as the contexts of entity mentions into a shared box space [30], where entity mentions are classified. In contrast to traditional classification methods that rely on vector spaces, high-dimensional box space leverages spatial relationships-such as separation, intersection, or overlap to classify entity mentions [18, 27]. High-dimensional vector spaces are foundational for semantic models (e.g., word embeddings). They capture nuanced relationships between concepts, supporting tasks like analogical reasoning and compositional thought.

While UFET methods based on box embeddings offer the advantages mentioned above, there are still some shortcomings to this approach. First, the influence of entity types is not considered or insufficiently considered when conditional probabilities are used to classify entity mentions [18, 27]. Second, gradient-based optimization methods tend to perform poorly in the case two boxes are either completely separated or overlap. To resolve such issue, BOX4Types [18] employs the Gumbel approximation, however it is an approximation method and cannot be computed exactly.

Similarly, to solve the problem, Qin et al. [27] proposed a distance-based approach, but their approach was too strict. To better resolve the aforementioned issue, this work introduces Entity Type-aware Ultra-fine Entity Typing with Adaptive Distance Optimization (ETUT), designed to overcome the identified limitations with little extra computational cost. The ETUT proposed in this paper contains two two modules, one of which is an adaptive module based on IoU and the other is an adaptive classification optimization module based on distance.

To tackle the first limitation, earlier research [4, 17, 20, 22, 23, 26, 31] generally modeled UFET in vector space. In vector space, all entity types share the same dimensions, making it challenging to distinguish between them and to account for their effects during classification. Recent work [18, 27] has introduced high-dimensional box space for modeling UFET, allowing entity type impacts to be considered during classification, given their diverse sizes and shapes. This study stresses that it is unjustifiable to overlook the influence of entity types, as indicated by the first limitation. In box space, the sizes of boxes representing entity types are inconsistent, particularly in the case entity types span different granularities or depth of type hierarchy. An entity type of coarse-grained typically associated with a larger box than an entity type of fine-grained, as the latter are usually subcategories of coarse-grained ones. The analysis reveals that entity mentions typically intersect more extensively with coarse-grained ones than with fine-grained ones. To exemplify this phenomenon, this paper uses the entity type /person and its sub-entity type /person/actor to illustrate the concept of type hierarchy. The bounding box representing an entity mention might be entirely enclosed by the bounding box for the coarse-grained entity type /person, while may exhibit only a limited overlap with the box for the fine-grained sub-type /person/actor, a specific type under /person. Consequently, entity mentions are typically classified into the general entity type /person with ease, but accurately identifying them as the fine-grained entity type /person/actor remains a more complex task. Nevertheless, the latter deliver more detailed and critical information, enhancing their importance in downstream applications [2, 15]. To tackle the aforementioned limitation, this paper presents an adaptive IoU-based module that improves model performance.

To alleviate the second limitation, the Gumbel approximation [32] is employed in recent work [18], partially mitigating this issue. The Gumbel approximation, being an approximate method, cannot be computed exactly, resulting in errors during training. While solutions for the second limitation are scarce in NLP [33], there is a wealth of approaches in computer vision, especially in object detection [34-40], such as GIoU [34], distance-IoU [35], CIoU [35], EIoU [39]. To resolve the second limitation of previous work, this work presents an adaptive distance-based optimization module, leveraging insights from predicted box refinement in computer vision.

The major contributions of this paper are as follows:

In this paper, an adaptive IoU-based module is introduced to integrate the effects of entity types into the classification process adaptively.

This work presents an adaptive distance-based module aimed at refining model performance by effectively handling scenarios where two boxes exhibit complete separation or full overlap.

Empirical evaluations conducted on the UFET and FGET datasets indicate that ETUT consistently out-performs baseline models, achieving state-of-the-art (SOTA) results.

The structure of this paper is as follows. Section 2 provides an overview of related work, Section 3 outlines the ETUT in detail, Section 4 reports experiments, and at the end Section 5summarizes the key findings of this paper.

2 RELATED WORK

In FGET as well as UFET, the number of entity types can range from hundreds [14] to thousands [19]. This substantial number of entity types in FGET and UFET introduces two primary challenges. In this section, we review existing work related to fine-grained and ultra-fine entity typing (FGET and UFET), focusing on three key aspects. Section 2.1 discusses the role and modeling of type hierarchies in entity typing. Section 2.2 reviews approaches specifically designed for ultra-fine entity typing. Section 2.3 summarizes prior efforts in addressing label noise issues, which are particularly prevalent in large-scale entity typing datasets. These subsections collectively provide the background and motivation for the proposed ETUT framework.

2.1 Type Hierarchy in FGET

Entity types in FGET as well as UFET typically organizes entity types into a hierarchical structure [20-22]. For in- stance, the OntoNotes dataset organizes its entity types into a three-level hierarchical structure, whereas the BBN dataset and FIGER dataset use a two-level hierarchical structure to organize their entity types. Earlier work on FGET has not adequately considered the significance of type hierarchical relationship among entity types, overlooking the type hierarchy and assuming all entity types to be of equal importance [12, 15]. In UFET, the type hierarchy serves as a fundamental mechanism that effectively boosts the model's classification capabilities by leveraging structured relationships among entity types [18, 22, 31].

One of the most foundational approaches in FGET is the NFETC method, introduced by Shimaoka et al. [17]. Shimaoka et al. [17] introduced a method to encode entity types, capturing the hierarchical relationships between them based on the type hierarchy. By leveraging the encoding strategy proposed by Shimaoka et al. [17], the entity types located in the same hierarchy can share parameters with their common parent, which enhances model performance, particularly for entity types that suffer from insufficient training data. Chen et al. [22] emphasized that penalties should vary across entity types at different hierarchical levels and introduced a ranking loss and a sub-entity type relation constraint loss. According to Xu et al. [23], treating entity types equally in FGET is not suitable, as it neglects the varying levels of specificity and importance among different entity types. Consider the case where an entity mention is classified as /person/athlete. In this case, it is more reasonable to assign /person as its entity type rather than /location, given that /person/athlete is inherently associated with /person. To address this, Xu et al. [23] introduced a hierarchy-aware loss function that mitigates the penalties for entity types that are more relevant or closely linked within the type hierarchy. To better capture the co-occurrence dependencies among entity types, Xiong et al. [41]

leveraged convolutional networks to model entity type co-occurrence statistics.

2.2 UFET

All of the above studies need to provide a well-defined type hierarchy to be effective, however, a well-defined type hierarchy is not provided in UFET.

To model hierarchical relationship without requiring explicit type hierarchy, López et al. [25] utilized hyperbolic space for FGET and UFET. Representations of both entity mentions and entity types are embedded in the shared hyperbolic space, with all relevant computations and operations executed within this space. To address the challenge of modeling hierarchical relationships without a type hierarchy as well as address the limitations of earlier work, Onoe et al. [18] embedded both entity mentions and entity types into a shared box space. Compared to hyperbolic space, high-dimensional box space provides clearer mathematical interpretation and is more effective at capturing entity type relationships. Despite its advantages, the box space-based UFET method overlooks the influence of entity types as well as lacks precise optimization when two boxes are either entirely overlapped or fully separated. In order to solve the above deficiencies of the existing high-dimensional box space, this paper proposes ETUT, which can effectively solve the limitations of the existing high-dimensional box space.

2.3 Label Noise in FGET

The vast number of entity types in FGET and UFET creates substantial difficulties for data annotation, which is why distant supervision [42] is often employed in the annotation process [14, 18]. However, distant supervision fails to account for the context of the entity mention, resulting in label noise for FGET and UFET [2, 23]. Due to the invisible nature of noise in the FGET and UFET annotation process, it is difficult to apply effective measures to address it. This challenge leads to a relatively small focus on data noise reduction in FGET and UFET studies, however there are some more classical data noise reduction methods in FGET [43, 44] and UFET [45].

Xin et al. [16] proposed a fine-grained entity typing (FGET) method enhanced by a large language model, leveraging the model to quantify the relationship between context and predicted entity types. Following the approach of Xin et al. [16], predicted entity types exhibiting stronger relevance to the context should experience reduced penalties. Zhang et al. [44] handled both clean data as well as noisy data in the same way, and introduced probabilistic-based approach to handle label noisy issue. Ren et al. [1] introduced a label noise processing approach that leverages PLL (partial-label loss), demonstrating considerable success in FGET. Xu et al. [23] highlighted the issue of out-of-context noise in FGET and introduced an enhanced PLL to mitigate the issue of out-of-context noise. Ren et al. [12] developed a method named AFET, which distinguishes between clean and noisy data by using PLL to manage noisy labels. Nevertheless, it requires an external KB (knowledge base) to function effectively. Pan et al. [45] introduced an approach utilizing Gaussian distribution that automatically eliminates the effects generated by noisy data in FGET and UFET. Onoe et al. [43] proposed an approach to mitigate label noise through filter and relabel modules, which utilizes a filter module to discard excessively noisy labels and are label module to reassign accurate labels to the filtered data. Chen et al. [21] presented clustering-driven FGET approach that separates dataset into noisy or clean subsets, using clean subset for model training as well as noisy subset for model optimization. Wu et al. [20] introduced a content- driven strategy that leverages random walk to eliminate the influence of noisy labels in FGET.

2.4 Summary

To highlight the distinctions between ETUT and prior research, it is important to note the following:

Unlike existing box-based methods [18, 27] that either overlook the influence of entity types or apply static treatment, ETUT introduces an adaptive IoU-based module that dynamically integrates the impact of entity types based on their spatial characteristics in box space.

In contrast to BOX4Types [18], which relies on Gumbel approximations for optimization when boxes are fully overlapped or separated, ETUT incorporates an adaptive distance module inspired by object detection techniques in computer vision, enabling precise and robust optimization even in these extreme cases.

Furthermore, ETUT employs a multi-objective optimization framework, jointly optimizing classification, adaptive IoU, and distance constraints, which previous work did not explore. These distinctions demonstrate ETUT's capacity to address fundamental challenges in both fine-grained and ultra-fine entity typing tasks.

3 METHODOLOGY

In this section, a comprehensive overview of the ETUT method proposed in this paper is presented. This section is structured as follows. The first part describes the problem statement, followed by the description of the input representation. The subsequent section presents a comprehensive description of the two modules proposed in this paper. At the final of this section, this paper introduce the overall objective function.

3.1 Problem Statement

This paper defines both FGET and UFET as multi-label classification problems, in which an entity mention m can be assigned to one or more entity types. In FGET, an entity mention can be classified into distinct entity types depending on the context in which it appears, and a sentence may contain multiple entity mentions, each potentially belonging to different entity types. Consequently, for a given entity mention m and its surrounding context s (the sentence in which m appears), FGET strives to classify m into one or multiple entity types, considering both m and s. If the probability of entity mention m belonging to entity type t^k exceeds

0.5, t^k will be assigned to m, where t^k □ T, T denotes the set of all entity types.

3.2 Input Representation

In line with prior FGET and UFET approaches, this paper utilizes entity mention m as well as its context s to construct the feature representation for FGET and UFET [18, 44, 46]. In contrast to previous FGET and UFET approaches that rely on traditional vector spaces, our method projects the representations of entity types as well as the representations of entity mentions into a shared box space, where the classification of entity mentions into their respective entity types occurs.

For an entity mention m and context s of entity mention m, the corresponding representation of entity mention m is constructed based on the pair (m, s). The above process can be mathematically expressed as:

$$x_{mm} = CLS_S \, SEP_m \tag{1}$$

In Eq. (1), the *CLS* token is specifically employed to encode the semantic representation of the entire expression, while the *SEP* token acts as a separator to delineate two different phrases.

Once the entity mention m is encoded as x_{mm}, a systematic approach is undertaken to project it into high-dimensional box space. The first step involves segmenting x_m into individual tokens, which serves as the foundation for subsequent operations in the model. The next step involves transforming the tokens of x_{mm} into the vector representation $h^{[CLS]} \in R^l$ by applying the [CLS] token from the encoder, such as BERT [47], where l signifies the dimension of the encoder's output space. At the end, the vector $h^{[CLS]}$ is mapped to $x \in R^{2d}$ via a highway network, where R^d denotes the coordinates of center point, and R^d denotes the offsets relative to center point.

In this work, box x is represented using two vectors: one is the center coordinates $\in R^d$, and another one is the offsets coordinates $\in R^d$ which is measured relative to c_x. As a result, the minimum coordinates as well as the maximum point coordinates of box is mathematically expressed as:

$$x_M\left(x_m\right) = \sigma\left(c_x - / + softplus\left(o_x\right)\right) \tag{2}$$

In Eq. (2), σ is an activation function, softplus within Eq. (2) is mathematically expressed as:

$$softplus\left(x\right) = \ln\left(1 + \mathrm{e}^x\right) \tag{3}$$

After the representation of entity mention m in the high-dimensional box space according to Eq. (1), the following step requires the projection of entity type t^k into the shared high-dimensional box space. In addition, the entity type t^k is mapped to its corresponding box representation y^k through a structured conversion mechanism, as introduced in [18]. The box embedding y^k is parameterized by its center coordinates $\in R^d$, an offset vector $\in R^d$ that defines its spread in high-dimensional space. Therefore, its boundary coordinates as well as is mathematically expressed as:

$$y_m^k \big/ y_M^k = \sigma\left(c_{y^k} - / + softplus\left(o_{y^k}\right)\right) \tag{4}$$

In box space, the entity mention m is encapsulated within a box whose volume is denoted as $Vol\left(x\right)$, while the entity type t^k is mapped to another box with volume $Vol\left(y^k\right)$, both serving as fundamental components of the embedding model.

Given the learned probability distributions for both the entity mention and entity type, the classification process involves computing their joint distribution, where the intersection between box x and box y^k serves as a probabilistic measure of entity mention m belonging to entity type y^k [18].

In high-dimensional box space, the probabilistic relationship between an entity mention and an entity type is modeled through the intersection of box x and box y^k, denoted as $x \cap y^k$. The intersection volume provides a measure of their compatibility, which can be mathematically formulated as follows:

$$\begin{aligned} &Vol\left(x \cap y^k\right) = \\ &\prod_i \max\left(0, \min\left(y_{M,i}^k, x_{M,i}\right) - \max\left(y_{m,i}^k, x_{m,i}\right)\right) \end{aligned} \tag{5}$$

In the equation Eq. (5), i refers to the i-th dimension of the high-dimensional box space, which contributes to the computation of the overall intersection volume through independent calculations along each axis.

Once the intersection volume between box x and box y^k has been computed, the subsequent step is to calculate the conditional probability that the entity mention m is associated with the entity type y^k. The conditional probability is computed as the percentage of to $Vol\left(x\right)$, which can be defined as:

$$p_\theta(t^k \mid \left(m,s\right)) = \frac{Vol\left(y^k \cap x\right)}{Vol\left(x\right)} \tag{6}$$

In Eq. (6), if $p_\theta\left(t^k\right)$ greater the threshold (set to 0.5 in this paper), t^k is designated as the predicted entity type for entity mention m. By leveraging this manner, the classification of an entity mention to its corresponding entity type is achieved through the relationship between y^k and x. Once the conditional probability is calculated through Eq. (6), the BCE loss function is applied to optimization the model:

$$\begin{aligned} L_c = &-\sum_{(m,s)\in D} \sum_l t_{real}^l \log(p_\theta(t^l \mid \left(m,s\right)) + \\ &+(1 - t_{real}^l)\log(1 - p_\theta(t^l \mid \left(m,s\right))) \end{aligned} \tag{7}$$

In Eq. (7), is equal to 0 or 1 to represent whether t^l is the real entity type of entity mention m, D represents the whole data.

Up to this stage, the paper has explained how to the model the FGET as well as UFET within the framework of box space in detail. The following section details how the adaptive IoU module incorporates the influence of entity types and how the distance-based module works in box space.

3.3 The Adaptive IoU Module

In this work, an adaptive IoU module is proposed to account for the influence of entity types within the context of FGET and UFET. Due to the fact that in box space, entity types located in varying hierarchical levels or entity types with different granularities correspond to different sizes of boxes (entity types close to the root node or coarse-grained entity types correspond to larger boxes, and entity types farther away from the root node or fine-grained entity types correspond to smaller boxes, this phenomenon is caused by FGETs and UFETs themselves). The adaptive IoU module discussed in this paper allows the model to self-adjust and learn how different entity types influence its output, thus strengthening the model's effectiveness in handling fine-grained entity types. In the context of FGET and UFET, where fine-grained entity types contain more detailed and diverse information, they are essential to improving the overall performance of the model.

In order to achieve the above objective, this paper first considers a traditional IoU module, which can be represented as follows:

$$p_{IoU} = \frac{Vol\left(y^k \cap x\right)}{Vol\left(y^k\right) + Vol\left(x\right) - Vol\left(y^k \cap x\right)} \tag{8}$$

Although Eq. (8) explores the integration of entity types in box space via an adaptive IoU-based module, it is not in a way designed to take into account the relationship between the box associated with an entity type as well as the box associated with an entity mention. In FGET and UFET, an entity type tends to have many entity mention corresponding to it. This results in the box linked to the entity type typically being larger than the box linked to the entity mention. Consequently, Eq. (8) causes a stronger influence from the entity type and a weaker influence from the entity mention. To mitigate the limitations of Eq. (8), this study presents a redesigned adaptive IoU module, as described below:

$$p_{\alpha IoU} = \frac{Vol\left(y^k \cap x\right)}{Vol\left(y^k\right) + Vol\left(x\right) - Vol\left(y^k \cap x\right)} \tag{9}$$

As shown in Eq. (9), the influence of entity type is introduced in box space skillfully.

In comparison with Eq. (6), Eq. (9) incorporates the influence of the box corresponding to t^k through the addition of $\alpha Vol\left(y^k\right)$. In order to restrict the output to the range [0, 1], $Vol\left(y^k \cap x\right)$ is incorporated into the denominator in Eq. (9). Therefore, Eq. (7) is employed to calculate the value associated with Eq. (9), and its value is defined as $L_{\alpha IoU}$.

It can be seen from Eq. (9), Eq. (9) tends to yield larger values for fine-grained entity types, which is in line with the objective of fine-grained entity types delivering more comprehensive information for downstream applications. The above analysis also fully demonstrates the theoretical justification of Eq. (9).

3.4 The Adaptive Distance Module

The adaptive distance module proposed in this paper is used to optimize two boxes exactly when they are completely overlapped or separated. It is worth to be noticed that the adaptive distance module proposed in this paper will make the two boxes to be close to each other within a certain range, instead of making the distance between the two boxes is 0.

The proposed adaptive distance module in this paper firstly needs to calculate the relative distance between the centroids of the two boxes, which is mathematically formulated as:

$$d_r = \frac{d_c\left(c_x, c_{y^k}\right)}{d_v\left(y^k \cup x\right)} \tag{10}$$

In Eq. (10), $d_c\left(c_x, c_{y^k}\right)$ denotes the distance between the centroids of the two boxes, $d_v\left(y^k \cup x\right)$ denotes the distance between the two most distant vertices of the two boxes. The purpose of this is to ensure that the distance is not too large or too small.

The above formula does not carry out the corresponding optimization when the distance is calculated, which could result in the distance between the two boxes converging to 0. To address the limitations of the aforementioned Eq. (10), this paper proposes an adaptive distance module, which can be specifically described as:

$$d_{rr} = \begin{cases} d_r - \beta, & \text{if } \ d_r \geq \beta \\ 0, & \text{if } \ d_r < \beta \end{cases} \tag{11}$$

where β is a threshold value used to modulate the distance between the two boxes.

For computational convenience, this paper denotes the loss generated by the proposed adaptive distance module as P_d, which can be expressed as:

$$P_d = 1 - d_{rr} \tag{12}$$

Accordingly, Eq. (7) can be employed to compute the value associated with Eq. (10), expressed by $L_{\beta D}$.

3.5 Multi-Objective Optimization Function

In order to combine the adaptive IoU module as well as the adaptive distance module proposed in this paper, both modules are used as optimization objectives for joint optimization with the classification module in this paper. In order to achieve the above objective, the optimization objective of this paper can be expressed as:

$$L_{\text{final}} = L_c + \lambda L_{\alpha IoU} + \eta L_{\beta D} \tag{13}$$

where λ and η in Eq. (13) are two tunable hyperparameters designed to balance the relative contributions of various modules.

Through the multi-objective optimization function of Eq. (13), the two modules proposed in this paper are effectively integrated into the classification model while maintaining the computational efficiency.

4 EVALUATION

This section validates the effectiveness of ETUT through comparisons with SOTA baselines on the FGET and UFET datasets. Additionally, ablation studies are conducted to evaluate the impact of each individual module proposed in this paper.

4.1 Datasets and Evaluation Metrics
4.1.1 Datasets

This work validates the proposed approach on commonly used FGET dataset and UFET dataset, including the Ultra-Fine Entity Typing (UFET) as well as the OntoNotes dataset (FGET). To maintain fairness in comparisons, the training, development, and test set divisions are consistent with those in previous approaches [18, 48].

UFET. Focuses on ultra-fine entity typing, predicting 10k+ free-form phrases (e.g., "skyscraper") for mentions in text. Uses head-word distant supervision and crowdsourced validation for high-precision, open-domain semantic understanding.

OntoNotes. A multi-task corpus for NER, POS tagging, and parsing. Annotates traditional entities (e.g., PERSON, LOC) with structured labels, serving as a benchmark for sequence labeling and syntactic tasks across genres.

4.1.2 Evaluation Metrics

As outlined in [18, 48], distinct evaluation metrics are applied based on the different dataset. For evaluating the UFET dataset[1], macro-precision, macro-recall, as well as macro-F1 are employed. Comparisons are conducted with baseline models on both the development and test sets [19, 41, 48] for the UFET dataset. Accuracy, macro-F1, and micro-F1 are adopted as the evaluation metrics for the OntoNotes dataset [25, 48, 49].

The formal definitions of Micro F1, Micro F1 and Accuracy are given as bellow.

$$Accuracy = \frac{1}{N}\sum_{i=1}^{N} I\left(T_i^{'} = T_i\right) \tag{14}$$

In Eq. (14), N denotes the total data, $T_i^{'}$ denotes the predicted labels, T_i denotes the real labels. And $I(x)$ is an indicator function used to determine whether two expressions are equal, which can be formally expressed as:

$$I(x) = \begin{cases} 1, & \text{if } T_i^{'} = T_i \\ 0, & \text{othrerwise} \end{cases} \tag{15}$$

Macro F_1 as well as Micro F_1 are variations of F_1, both representing the harmonic average of recall and precision, with the key difference that precision and recall are calculated differently. And the calculation of F_1 can be formalized as:

$$F_1 = \frac{2 * recall * precision}{recall + precision} \tag{16}$$

The recall and precision in macro F_1 can be mathematically formulated as:

$$recall = \frac{1}{N}\sum_{i=1}^{N} \frac{\left|T_i^{'} \cap T_i\right|}{T_i} \tag{17}$$

$$precision = \frac{1}{N}\sum_{i=1}^{N} \frac{\left|T_i^{'} \cap T_i\right|}{T_i} \tag{18}$$

And the recall and precision in micro F_1 can be mathematically formulated as:

$$recall = \frac{\sum_{i=1}^{N}\left|T_i^{'} \cap T_i\right|}{\sum_{i=1}^{T}\left|T_i^{'}\right|} \tag{19}$$

$$recall = \frac{\sum_{i=1}^{N}\left|T_i^{'} \cap T_i\right|}{\sum_{i=1}^{T}\left|T_i^{'}\right|} \tag{20}$$

4.2 Comparison of Baselines and Hyperparameter Settings
4.2.1 Comparison of Baselines

In this work, ETUT is evaluated on the FGET and UFET dataset by benchmarking it against the following baselines:

LabelGCN [4]. LabelGCN incorporates a convolutional layer into the output of the LSTM to model potential relationships among entity types.

NFETC-CLHLS [17]. Xu et al. emphasize that "clean" datasets used in existing FGET are not entirely noise-free. Their solution involves employing a structured curriculum learning approach in three stages to refine FGET model training.

[1] https://github.com/nlpAThits/hyfiAThits/hyfi

LRN [49]. LRN adopts inductive and deductive reasoning for intrinsic and extrinsic dependencies among entity types. It uses an LSTM-based auto-regressive network to model extrinsic dependencies and a bipartite attribute graph to capture intrinsic ones.

NFETC [23]. NFETC is an essential method in FGET, offering a novel feature representation for entity mentions. By combining partial label loss with hierarchy-aware loss, it efficiently captures the type hierarchy in FGET.

UFET [19]. The UFET model, which first introduced the UFET dataset, utilizes LSTM as well as convolutional neural network. The assignment of entity mentions to entity types is determined by the inner product of their representations. At the end, a modified objective function is used to address entity type imbalance.

NFETC-SSL [43]. Xu et al. identify persistent challenges in FGET datasets, including confirmation bias in noisy data and false positives in clean data, de- spite previous segmentation efforts. Their proposed semi-supervised solution incorporates mixed label smoothing and pseudo-labeling for improved accuracy.

BOX4Types [18]. BOX4Types captures relationships between entity types in box space, even when no type hierarchy is available. BOX4Types leverages bert-large as an encoder for feature extraction. For comparison with other types of methods, BOX4Types implements two methods for FGET: BOX4Types-box as well as BOX4Types-vector.

HY XLarge [22]. HY XLarge employs a hyperbolic model to infer the latent type hierarchy and classify entity mentions into entity types within hyperbolic space.

Onoe et al. (ELMo**)** [18]. Onoe et al. [43] introduced an approach to mitigate label noise by employing filter and relabel modules. The filter module discards excessively noisy labels, while the relabel module reassigns accurate labels to the filtered data.

NPCRF [11]. To effectively explore and capture the intricate relationships between entity types, NPCRF utilizes an advanced approach known as the neural pairwise conditional random field, which enhances the modeling of interactions and dependencies among entity types.

The baseline models selected in this work span a broad spectrum of approaches in fine-grained and ultra-fine entity typing. These include:

- Traditional vector-based methods (e.g., NFETC, UFET-biLSTM), which serve as foundational models for FGET and UFET.
- Label noise robust models (e.g., NFETC-CLHLS, NFETC-SSL, Onoe et al. 2019), which are relevant given the noisy annotation settings in UFET datasets.
- Hierarchy-aware or structure-enhanced methods (e.g., LRN, Chen et al. 2022), which utilize type dependency or structural information to improve classification, aligning with ETUT's attention to entity type relations.
- Box embedding-based methods (e.g., BOX4Types), which are the most directly comparable to ETUT, as they also operate in high-dimensional box space. ETUT is designed as an improvement over such methods by introducing adaptive IoU and distance modules.
- Hyperbolic models (e.g., HY XLarge), which aim to capture latent type hierarchies without requiring explicit structure - similar in motivation to ETUT, but differing in representation space.

By including these representative baselines, we aim to ensure that ETUT is evaluated against a diverse and comprehensive set of relevant models, covering multiple perspectives on the FGET and UFET tasks.

4.2.2 Hyperparameter Settings

Due to the fact that the number of entity types varies significantly across different datasets, the influence of the two proposed modules are not uniform. Therefore, hyper-parameters are used to control their influence on different datasets. In order to enable researchers to have a quick understanding of the hyperparameters of this paper, this paper summarizes the hyperparameters used in this paper, the specific hyperparameters for different dataset are shown in Tab. 1.

Table 1 Hyperparameter settings

Hyperparameters	UFET	original OntoNotes
α	0.02	10
β	0.0005	0.002
λ	0.002	0.005
η	0.01	0.0005
Batch size	32	32

This work utilizes roberta-base as the encoder for feature extraction, and utilizes the Adam optimizer for model optimization. The hyperparameters are tuned artificially one by one in this work, rather than being selected via hyperparameter search as previous work [18, 45].

The hyperparameter settings for the ETUT model are carefully adjusted based on the characteristics of each dataset. Specifically, α and β control the contribution of the adaptive IoU and distance modules, respectively. Since the UFET dataset lacks an explicit type hierarchy and contains a large number of fine-grained labels, we use relatively smaller values for α and β (0.02 and 0.0005) to prevent these modules from dominating the optimization process. Conversely, for the OntoNotes dataset, which includes a clearer hierarchical structure and more balanced label distribution, higher values for α and β (10 and 0.002) allow the model to make fuller use of the structural information and distance-based refinement.

We adopt a manual tuning strategy instead of grid or automated search, aiming to balance performance with training stability. The tuning process is guided by development set performance and prior findings in related work such as BOX4Types [18] and Pan et al. [45], ensuring fairness in comparison while aligning with practical implementation needs.

4.3 Experimental Results on Various Datasets

4.3.1 Experimental Results on the UFET Test Set and Development Set

a) Experimental results on the UFET test set

In Tab. 2, the macro precision, macro recall, and macro F1 scores for the UFET test set are presented, offering a comparative performance overview of various approaches. The experimental results on the UFET test set shown in Tab. 2 demonstrates that ETUT achieves the highest

Table 2 Macro precision/recall/F1 results on the UFET test set

Model	Macro precision	Macro recall	Macro F1
Label GCN	50.3	29.2	36.9
UFET-biLSTM	47.1	24.2	32.0
BOX4Types (vector)	53.0	36.3	43.1
BOX4Types (box)	52.8	38.8	44.8
LRN	54.5	38.9	45.4
Onoe et al. (2019) (ELMo)	51.5	33.0	40.2
* NPCRF	50.8	43.0	46.6
ETUT	**62.2**	**39.4**	**48.3**

The best results have been clearly emphasized using bold to distinguish them from other values.

A few publications did not provide the names of their methodology, so in this work, the methods are represented by the format "author name (year)" to clarify the source of each method.

"*" indicates that results are recapitulated using the primal parameters provided in their publication.

Table 3 Macro precision/recall/F1 results on the UFET development set

Model	Macro precision	Macro recall	Macro F1
LRN	53.7	38.6	44.9
UFET-biLSTM	48.1	23.2	31.3
HY XLarge	43.4	34.2	38.2
t Label GCN	49.3	28.1	35.8
BOX4Types (box)	52.9	39.1	45.0
BOX4Types (vector)	53.3	36.7	43.5
* NPCRF	51.1	42.4	46.4
Onoe et al. (2019) (BERT)	51.6	32.8	40.1
Onoe et al. (2019) (ELMo)	50.7	33.1	40.1
ETUT	**60.9**	**40.1**	**48.4**

The *t* symbol signifies that the data are derived from the paper of López et al. [25, 48].

Macro-F1 score among all approaches on the UFET test set. ETUT demonstrates a substantial enhancement in terms of macro recall as well as macro precision when compared to other approaches. When compared to box-based UFET approaches, ETUT demonstrates notable enhancement in terms of macro recall as well as macro precision, ultimately achieving superior macro F1 performance over BOX4Types (box). The above experimental results indicates that ETUT effectively addresses the inherent limitations of box-based UFET approaches as well as box-based FGET approaches, which also emphasizes the usefulness of the ETUT pre- sented in this work.

b) Experimental results on the UFET development set

As for the UFET dataset, initially introduced by Choi et al. [19], was constructed using a combination of manual annotation, entity linking, and head-word extraction techniques to ensure data quality. To comprehensively validate the influence of individual techniques on data quality, Choi et al. [19] employed the UFET development set for rigorous evaluation. Following this framework, later research [18, 25] adhered to this approach to maintain consistency in evaluation and facilitate meaningful comparisons. In order to ensure fairness and validity, this paper incorporates results from the development set of UFET, enabling a comprehensive evaluation of experimental performance. In Tab. 3, a thorough examination of the experimental results on the development set of UFET is displayed.

The experimental results in Tab. 3 clearly illustrate that ETUT achieves superior performance over all baseline methods, as evidenced by its higher macro F1 score. ETUT achieves notable improvements in both macro-recall and macro-precision compared to other methods, highlighting its overall performance and effectiveness.

The experimental results from both the development set and the test set of UFET clearly show that the ETUT achieves superior performance, which strongly supports the effectiveness of ETUT on the UFET dataset.

4.3.2 Experimental Results on the OntoNotes Dataset

The performance of ETUT on the UFET dataset has been verified in previous experiments. The following experiments will focus on evaluating the performance ETUT on other FGET datasets, such as the OntoNotes dataset.

Tab. 4 summarizes the results of the experiments for the OntoNotes dataset. As shown in Tab. 4, ETUT demonstrates superior performance, achieving the highest micro F1 as well as macro F1. Notably, ETUT presented in this paper achieves remarkable improvements in micro F1 as well as macro F1, surpassing latest method. The above experimental results on UFET as well as OntoNotes indicate ETUT works well across datasets regardless of whether the dataset has an explicit type hierarchy or not.

Table 4 Accuracy, Macro F1 and Micro F1 results on the OntoNotes dataset

Model	Accuracy	Macro F1	Micro F1
LRN	56.6	77.6	71.8
AFET	55.1	71.1	64.7
Chen et al. (2020)	58.7	73.0	68.1
Chen et al. (2022)	59.2	76.5	71.0
BOX4Types (box)	-	77.3	70.9
BOX4Types (vector)	-	76.2	68.9
Onoe et al. (2019) (BERT)	51.8	76.6	69.1
ETUT	**59.6**	**80.6**	**72.6**

4.4 Ablation Study on the UFET Dataset and OntoNotes Dataset

4.4.1 UFET Dataset Ablation Study

To evaluate the individual contributions of each module to the overall effectiveness of our approach, this paper conducts comprehensive ablation studies on two datasets. These experiments systematically remove specific components to analyze their influence on the behavior of the presented methodology.

Tab. 5 presents the experimental results of the ablation study on the UFET dataset, which provide a comprehensively analysis of the distinct contributions of individual modules to the overall performance. The results from the experiments indicate that each proposed module contributes significantly to enhancing performance metrics when applied independently. Furthermore, the integration of the two modules not only enables smooth interaction but also capitalizes on their unique and complementary strengths, this greatly improves performance.

Table 5 UFET dataset ablation study

Model	Macro precision	Macro recall	Macro F1
ETUT w/o ($L\beta D + L\alpha IoU$)	63.9	36.6	46.5
ETUT w/o $L\beta D$	62.4	38.5	47.6
ETUT w/o $L\alpha IoU$	63.7	38.3	47.8
ETUT	**62.2**	39.4	**48.3**

In each row, we remove a single module. The term "w/o" indicates the deletion of the relevant module according to the ETUT model.

Table 6 OntoNotes dataset ablation study

Model	Accuracy	Macro F1	Micro F1
ETUT w/o ($L\beta D + L\alpha IoU$)	58.7	79.9	71.7
ETUT w/o $L\beta D$	59.1	80.3	72.4
ETUT w/o $L\alpha IoU$	59.4	80.4	72.1
ETUT	**59.6**	**80.6**	**72.6**

4.4.2 OntoNotes Dataset Ablation Study

Beyond the ablation study performed on the UFET dataset, this paper further investigates the proposed approach by performing ablation study on the OntoNotes dataset, which further verifies the robustness as well as generalizability of ETUT. Tab. 6 shows specifically the results of the ablation study on the OntoNotes dataset.

The results presented in Tab. 6, provide a comprehensive assessment of the impact of each module on our methodology. Experimental results suggest that ETUT attains a well-balanced performance between accuracy, macro F1 and micro F1, underscoring its ability to perform consistently well across different performance metrics.

5 CONCLUSION

This paper introduces Entity Type-aware Ultra-fine Entity Typing with Adaptive Distance Optimization (ETUT), which is a new methodology that addresses key challenges in fine-grained entity typing as well as ultra-fine entity typing. ETUT method incorporates two adaptive modules: one that senses the entity type, and another that enhances model training, and both can lead to improved performance. The effectiveness of these modules is verified by the results of the experiments presented in this paper.

In future work, we plan to explore several promising directions. First, we aim to further enhance the adaptive modules by incorporating external knowledge sources, such as structured knowledge graphs, to refine entity type representations. Second, extending ETUT to multilingual or cross-lingual entity typing tasks could significantly broaden its applicability. Third, we are interested in investigating how ETUT can be integrated with large language models (LLMs) to improve zero-shot or few-shot entity typing. Finally, future research could explore how to automatically learn or refine latent type hierarchies from data, making the model more interpretable and adaptive to new domains.

Acknowledgments

The works described in this paper are supported by (1) The Henan provincial Science and Technology Research Project (252102210089): Research on Knowledge Tracking Based on Fine-Grained Educational Knowledge Graph. (2) Henan Province Undergraduate Universities Research-Oriented Teaching Series Project: A Study on the Transformation of Digital Intelligence-Driven Research-Based Teaching Methods - A Case Study of Local Undergraduate Institutions. (3) Henan Province Second Batch of Specialized Innovation Integration Characteristic Demonstration Courses: PHP Programming.

6 REFERENCES

[1] Ren, X., He, W., Qu, M., Voss, C. R., Ji, H., & Han, J. (2016). Label noise reduction in entity typing by heterogeneous partial-label embedding. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1825-1834. https://doi.org/10.1145/2939672.2939822

[2] Wang, R., Hou, F., Cahan, S. F., Chen, L., Jia, X., & Ji, W. (2022). Fine-grained entity typing with a type taxonomy: A systematic review. *IEEE Transactions on Knowledge and Data Engineering*, *35*(5), 4794-4812. https://doi.org/10.1109/tkde.2022.3148980

[3] Neelakantan, A. & Chang, M. W. (2015). Inferring missing entity type instances for knowledge base completion: New dataset and methods. *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, 515-525. https://doi.org/10.3115/v1/n15-1054

[4] Jin, H., Hou, L., Li, J., & Dong, T. (2019). Fine-grained entity typing via hierarchical multi graph convolutional networks. *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, 4969-4978. https://doi.org/10.18653/v1/d19-1502

[5] Recasens, M., De Marneffe, M. C., & Potts, C. (2013). The life and death of discourse entities: Identifying singleton mentions. *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, 627-633.

[6] Ling, X., Singh, S., & Weld, D. S. (2015). Design challenges for entity linking. *Transactions of the Association for Computational Linguistics*, *3*(1), 315-328. https://doi.org/10.1162/TACL_A_00141

[7] Onoe, Y. & Durrett, G. (2020). Fine-grained entity typing for domain independent entity linking. *Proceedings of the AAAI Conference on Artificial Intelligence*, 8576-8583. https://doi.org/10.1609/aaai.v34i05.6380

[8] Miwa, M. & Sasaki, Y. (2014). Modeling joint entity and relation extraction with table representation. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 1858-1869. https://doi.org/10.3115/v1/d14-1200

[9] Yaghoobzadeh, Y., Adel, H., & Schütze, H. (2016). Noise mitigation for neural entity typing and relation extraction. *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*, 1183-1194. https://doi.org/10.18653/v1/e17-1111

[10] Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., & Liu, Q. (2019). ERNIE: Enhanced language representation with informative entities. *Proceedings of the Annual Meeting on Association for Computational Linguistics*, 1441-1451. https://doi.org/10.18653/v1/p19-1139

[11] Yavuz, S., Gür, I., Su, Y., Srivatsa, M., & Yan, X. (2016). Improving semantic parsing via answer type inference. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 149-159. https://doi.org/10.18653/v1/d16-1015

[12] Ren, X., He, W., Qu, M., Huang, L., Ji, H., & Han, J. (2016). AFET: Automatic fine-grained entity typing by hierarchical partial-label embedding. *Proceedings of the Conference on*

Empirical Methods in Natural Language Processing, 1369-1378. https://doi.org/10.18653/v1/d16-1144

[13] Zhao, F., Gui, X., Huang, Y., Jin, H., & Yang, L. T. (2020). Dynamic entity-based named entity recognition under unconstrained tagging schemes. *IEEE Transactions on Big Data, 8*(4), 1059-1072. https://doi.org/10.1109/TBDATA.2020.2998770

[14] Ling, X. & Weld, D. (2012). Fine-grained entity recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 94-100. https://doi.org/10.1609/aaai.v26i1.8122

[15] Xin, J., Lin, Y., Liu, Z., & Sun, M. (2018). Improving neural fine-grained entity typing with knowledge attention. *Proceedings of the AAAI Conference on Artificial Intelligence*, 5997-6004. https://doi.org/10.1609/aaai.v32i1.12038

[16] Xin, J., Zhu, H., Han, X., Liu, Z., & Sun, M. (2018). Put it back: Entity typing with language model enhancement. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 993-998. https://doi.org/10.18653/v1/d18-1121

[17] Shimaoka, S., Stenetorp, P., Inui, K., & Riedel, S. (2016). Neural architectures for fine-grained entity type classification. *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*, 1271-1280. https://doi.org/10.18653/v1/e17-1119

[18] Onoe, Y., Boratko, M., McCallum, A., & Durrett, G. (2021). Modeling fine-grained entity types with box embeddings. *Proceedings of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing*, 2051-2064. https://doi.org/10.18653/v1/2021.acl-long.160

[19] Choi, E., Levy, O., Choi, Y., & Zettlemoyer, L. (2018). Ultra-fine entity typing. *Proceedings of the Annual Meeting on Association for Computational Linguistics*, 87-96. https://doi.org/10.18653/v1/p18-1009

[20] Wu, J., Zhang, R., Mao, Y., Guo, H., & Huai, J. (2019). Modeling noisy hierarchical types in fine-grained entity typing: A content-based weighting approach. *Proceedings of the International Joint Conference on Artificial Intelligence*, 5264-5270. https://doi.org/10.24963/IJCAI.2019/731

[21] Chen, B., Gu, X., Hu, Y., Tang, S., Hu, G., Zhuang, Y., & Ren, X. (2019). Improving distantly-supervised entity typing with compact latent space clustering. *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, 2862-2872. https://doi.org/10.18653/v1/n19-1294

[22] Chen, T., Chen, Y., & Van Durme, B. (2020). Hierarchical entity typing via multi-level learning to rank. *Proceedings of the Annual Meeting on Association for Computational Linguistics*, 8465-8475. https://doi.org/10.18653/v1/2020.acl-main.749

[23] Xu, P. & Barbosa, D. (2018). Neural fine-grained entity type classification with hierarchy-aware loss. *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, 16-25. https://doi.org/10.18653/v1/n18-1002

[24] Murty, S., Verga, P., Vilnis, L., Radovanovic, I., & McCallum, A. (2018). Hierarchical losses and new resources for fine-grained entity typing and linking. *Proceedings of the Annual Meeting on Association for Computational Linguistics*, 97-109. https://doi.org/10.18653/v1/p18-1010

[25] López, F. & Strube, M. (2020). A fully hyperbolic neural model for hierarchical multi-class classification. *Findings of the Association for Computational Linguistics*, 460-475. https://doi.org/10.18653/v1/2020.findings-emnlp.42

[26] Wu, J., Zhang, R., Mao, Y., Shahrbabak, M. S., & Huai, J. (2021). Hierarchical modeling of label dependency and label noise in fine-grained entity typing. *Proceedings of the International Joint Conference on Artificial Intelligence*, 3950-3956. https://doi.org/10.24963/IJCAI.2021/544

[27] Qin, Y., Wang, Y., Li, J., Mao, S., Wang, H., & Jiang, Y. (2023). Improved box embeddings for fine-grained entity typing. *IEEE Transactions on Big Data, 9*(6), 1631-1642. https://doi.org/10.1109/TBDATA.2023.3310239

[28] Nickel, M. & Kiela, D. (2017). Poincaré embeddings for learning hierarchical representations. *Proceedings of the International Conference on Neural Information Processing Systems*, 6341-6350.

[29] Ganea, O., Bécigneul, G., & Hofmann, T. (2018). Hyperbolic entailment cones for learning hierarchical embeddings. *Proceedings of International Conference on Machine Learning*, 1646-1655.

[30] Vilnis, L., Li, X., Murty, S., & McCallum, A. (2018). Probabilistic embedding of knowledge graphs with box lattice measures. *Proceedings of the Annual Meeting on Association for Computational Linguistics*, 263-272. https://doi.org/10.18653/v1/p18-1025

[31] Jin, H., Hou, L., Li, J., & Dong, T. (2018). Attributed and predictive entity embedding for fine-grained entity typing in knowledge bases. *Proceedings of the International Conference on Computational Linguistics*, 282-292.

[32] Dasgupta, S., Boratko, M., Zhang, D., Vilnis, L., Li, X., & McCallum, A. (2020). Improving local identifiability in probabilistic box embeddings. *Proceedings of the International Conference on Neural Information Processing Systems*, 182-192.

[33] Abboud, R., Ceylan, I., Lukasiewicz, T., & Salvatori, T. (2020). Boxe: A box embedding model for knowledge base completion. *Proceedings of the 34th International Conference on Neural Information Processing Systems*, 9649-9661.

[34] Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., & Savarese, S. (2019). Generalized intersection over union: A metric and a loss for bounding box regression. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 658-666. https://doi.org/10.1109/CVPR.2019.00075

[35] Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., & Ren, D. (2020). Distance-IoU loss: Faster and better learning for bounding box regression. *Proceedings of the AAAI Conference on Artificial Intelligence*, 12993-13000. https://doi.org/10.1609/aaai.v34i07.6999

[36] Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4: Optimal speed and accuracy of object detection.

[37] Zheng, Z., Wang, P., Ren, D., Liu, W., Ye, R., Hu, Q., & Zuo, W. (2021). Enhancing geometric factors in model learning and inference for object detection and instance segmentation. *IEEE Transactions on Cybernetics*, *52*(8), 8574-8586. https://doi.org/10.1109/TCYB.2021.3095305

[38] He, J., Erfani, S., Ma, X., Bailey, J., Chi, Y., & Hua, X. S. (2021). α-IoU: A family of power intersection over union losses for bounding box regression. *Proceedings of the International Conference on Neural Information Processing Systems*, 20230-20242.

[39] Zhang, Y. F., Ren, W., Zhang, Z., Jia, Z., Wang, L., & Tan, T. (2022). Focal and efficient IoU loss for accurate bounding box regression. *Neural Computing*, *506*, 146-157. https://doi.org/10.1016/j.neucom.2022.07.042

[40] Dong, C. & Duoqian, M. (2023). Control distance IoU and control distance IoU loss for better bounding box regression. *Pattern Recognition*, *137*, 109256. https://doi.org/10.1016/j.patcog.2022.109256

[41] Xiong, W., Wu, J., Lei, D., Yu, M., Chang, S., Guo, X., & Wang, W. Y. (2019). Imposing label-relational inductive bias for extremely fine-grained entity typing. *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, 773-784. https://doi.org/10.18653/v1/n19-1084

[42] Mintz, M., Bills, S., Snow, R., & Jurafsky, D. (2009, August). Distant supervision for relation extraction without

labeled data. *Proceedings of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing*, 1003-1011.
https://doi.org/10.1007/978-3-319-10888-9_35

[43] Onoe, Y. & Durrett, G. (2019). Learning to denoise distantly-labeled data for entity typing. *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, 2407-2417.
https://doi.org/10.18653/v1/n19-1250

[44] Zhang, H., Long, D., Xu, G., Zhu, M., Xie, P., Huang, F., & Wang, J. (2021). Learning with noise: Improving distantly-supervised fine-grained entity typing via automatic relabeling. *Proceedings of the International Joint Conference on Artificial Intelligence*, 3808-3815.
https://doi.org/10.24963/ijcai.2021/527

[45] Pan, W., Wei, W., & Zhu, F. (2022). Automatic noisy label correction for fine-grained entity typing. *Proceedings of the International Joint Conference on Artificial Intelligence*, 4317-4323.

[46] Lin, Y. & Ji, H. (2019). An attentive fine-grained entity typing model with latent type representation. *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, 6197-6202.
https://doi.org/10.18653/v1/d19-1641

[47] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, 4171-4186.
https://doi.org/10.18653/v1/n19-1423

[48] Liu, Q., Lin, H., Xiao, X., Han, X., Sun, L., & Wu, H. (2021). Fine-grained entity typing via label reasoning. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 4611-4622.
https://doi.org/10.18653/v1/2021.emnlp-main.378

[49] Chen, Y., Cheng, J., Jiang, H., Liu, L., Zhang, H., Shi, S., & Xu, R. (2022). Learning from sibling mentions with scalable graph inference in fine-grained entity typing. *Proceedings of the Annual Meeting on Association for Computational Linguistics*, 2076-2087.
https://doi.org/10.18653/v1/2022.acl-long.147

Contact information:

Feng WANG
(Corresponding author)
School of Computer Science and Technology, ZhouKou Normal University,
ZhouKou 466001, China
E-mail: wang7984@163.com

YiXiu QIN
School of Computer Science and Technology, ZhouKou Normal University,
ZhouKou 466001, China
E-mail: yixiu@m.scnu.edu.cn