

Article

Learning from Knowledge Graphs: Neural Fine-Grained Entity Typing with Copy-Generation Networks

Zongjian Yu ¹, Anxiang Zhang ², Huali Feng ¹, Huaming Du ¹ , Shaopeng Wei ¹ and Yu Zhao ^{1,*} 

¹ Financial Intelligence and Financial Engineering Key Laboratory of Sichuan Province, Southwestern University of Finance and Economics, Chengdu 611130, China; yuzj@swufe.edu.cn (Z.Y.); fenghuali0723@163.com (H.F.); dhmfcc@swufe.edu.cn (H.D.); 2191201z5003@swufe.edu.cn (S.W.)

² School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA; adamzhang1679@gmail.com

* Correspondence: zhaoyu@swufe.edu.cn

Abstract: Fine-grained entity typing (FET) aims to identify the semantic type of an entity in a plain text, which is a significant task for downstream natural language processing applications. However, most existing methods neglect rich known typing information about these entities in knowledge graphs. To address this issue, we take advantage of knowledge graphs to improve fine-grained entity typing through the use of a copy mechanism. Specifically, we propose a novel deep neural model called COPYFET for FET via a copy-generation mechanism. COPYFET can integrate two operations: (i) the regular way of making type inference from the whole type set in the generation model; (ii) the new copy mechanism which can identify the semantic type of a mention with reference to the type-copying vocabulary from a knowledge graph in the copy model. Despite its simplicity, this mechanism proves to be powerful since extensive experiments show that COPYFET outperforms state-of-the-art methods in FET on two benchmark datasets (FIGER (GOLD) and BBN). For example, COPYFET achieves the new state-of-the-art score of 76.4% and 83.6% on the accuracy metric in FIGER (GOLD) and BBN, respectively.

Keywords: knowledge graphs; fine-grained entity typing; copy-generation networks; cross-entropy



Citation: Yu, Z.; Zhang, A.; Feng, H.; Du, H.; Wei, S.; Zhao, Y. Learning from Knowledge Graphs: Neural Fine-Grained Entity Typing with Copy-Generation Networks. *Entropy* **2022**, *24*, 964. <https://doi.org/10.3390/e24070964>

Academic Editor: Kang Hao Cheong

Received: 20 June 2022

Accepted: 10 July 2022

Published: 11 July 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Fine-grained entity typing (FET) aims to infer the possible semantic type of an entity mention (i.e., a sequence of token spans representing an entity) [1,2]. Different from a traditional entity-typing task that typically classifies entities into coarse-grained types (e.g., *person*, *location*, *organization*) [3,4], FET aims to assign an entity with more specific types [5,6], which usually follow a hierarchical structure that can provide more semantic information about the entity [7,8], such as */person/politician*, */book/author*, etc. FET is a significant subtask of named-entity recognition (NER) [9] for downstream natural language processing (NLP) applications, such as relation extraction [10,11], question answering [12,13], knowledge base population [14], and recommendation [15,16].

In FET, knowledge graphs (KGs) usually play an important role. For example, given large-scale KGs, FET systems resort to distant supervision [10] to generate large training corpora [9,17,18] (i.e., to label entity mentions in the training corpus with all types associated with the entity in KGs). Although distant supervision can eliminate the high cost in labeling training data with KGs, how to efficiently encode a KG's typing knowledge into FET model is still underexplored.

In this paper, we concentrate on how to take advantage of KGs to improve FET in the process of type inference. In fact, the correct type information about a mention usually can be found in large-scale knowledge graphs, such as Freebase [19], YAGO [20], DBpedia [21], OntoNotes [22], and Few-nerd [23], which typically have already recorded a large number of entity-typing facts with various context from corpus. Consider the following example

from Wikipedia: “In 2006, *Obama* released *The Audacity of Hope* that expanded upon the themes in their convention speech.” The entity mention “*The Audacity of Hope*” can directly be predicted as a *book* by copying the type information of the known entity-typing fact (“*The Audacity of Hope*”, */things/book*) from KGs to the mention. For cases that require the understanding of the entity context, using a copy mechanism is also quite beneficial. In the previous example where “*Obama*” is the entity mention, copying all the types of “*Barack Obama*” in KGs (i.e., */people/person*, */book/author*, */person/politician*, etc.) as the priority typing candidates of the mention is still informative for inferring the correct type (i.e., */book/author*) that fits the context best, since they can substantially narrow down the optimal list of possible type labels (referred to as the type’s *copying vocabulary* from here onward). According to the statistics in Table 1, over 49% and 64% of all the manually annotated mentions’ typing facts in FIGER (GOLD) [9] and BBN [24] (testing data) have already been included in their existing KGs, respectively, which inspires us to improve FET by learning rich known entity-typing information from KGs.

Table 1. The percentage of the entity mentions’ typing facts that have been included in the existing knowledge graphs. The statistical analysis indicates they are over 49% and 64% in two benchmark datasets FIGER (GOLD) and BBN (testing data), respectively. These observations are the key motivation that inspired us to take advantage of knowledge graphs to improve fine-grained entity typing through the use of a **copy mechanism** in this paper.

Benchmark FET Datasets	# Type	# Testing Mentions	# Typing Facts Included in KG	KG Coverage
Wiki/FIGER (GOLD) [9]	128	563	280	49.73%
BBN [24]	56	13,282	8505	64.03%

To this end, we incorporate a copy mechanism in fine-grained entity typing, and propose a novel deep neural model called COPYFET for FET via a copy-generation mechanism. A copy mechanism can significantly avoid unnecessary mistakes and improve the accuracy in the type inference process. It is similar to the copy mechanism in the keyphrase generation [25] and abstractive summarization [26,27], which allows a language generator to copy items from the source text directly, in order to help generate reliable results that keep salient information from the source text.

Specifically, COPYFET includes two submodels of inference, i.e., a copy model and a generation model, as shown in Figure 1. COPYFET has two submodels: (i) the first one conducts the regular way of making type inference from the whole type set, i.e., generation model; (ii) the second one uses the new copy mechanism which can identify the semantic type of a mention with reference to the type’s *copying vocabulary* from KGs, called copy model. Both models are combined to build the final type inference model. Since it is unknown that a certain entity mention corresponds to a certain entity in KGs, we propose to perform entity linking as a solution to generate the type’s *copying vocabulary*.

Extensive experimental results on two benchmark FET datasets demonstrate that the proposed COPYFET can effectively conduct fine-grained entity typing by incorporating a copy model with a generation model in both training and inference. For example, COPYFET achieves 76.4% and 83.6% absolute strict accuracy on the benchmark datasets FIGER (GOLD) and BBN, respectively.

The contributions of this paper are threefold: (1) We propose to take advantage of knowledge graphs to improve fine-grained entity typing through the use of a copy mechanism. (2) Specifically, we propose COPYFET, a simple but effective neural fine-grained entity-typing model that incorporates a copy mechanism in FET via a copy-generation framework. (3) We conduct empirical experiments on two benchmark datasets, which demonstrate that incorporating a copy mechanism highlights the superiority of the proposed COPYFET over previous SOTA models in a FET task.

The structure of the paper is as follows. In Section 2, we provide a brief review of related works. In Section 3, we describe the methodology of our model. In Section 4,

we present the cross-entropy learning method. Section 5 presents experimental results followed by their discussion. Finally, Section 6 gives the conclusion and future directions of this research.

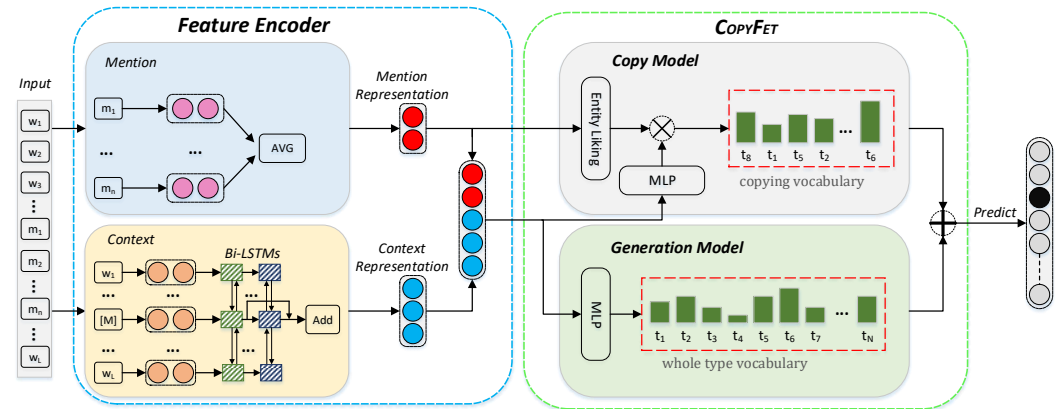


Figure 1. The overall framework of our proposed model COPYFET. In Feature Encoder (left box), it obtains the *mention representation* by averaging (Equation (2)) and the *context representation* by Bi-LSTM (Equation (3)). In COPYFET (right box), the green bar indicates the probability score calculated by the **copy model** (Section 3.1) and the **generation model** (Section 3.2). The copy model is able to learn to predict from a much more delimited candidate space, i.e., the type's *copying vocabulary*, than the whole type vocabulary, on which the generation model makes a prediction. The final type prediction agrees with both of them (Section 3.3).

2. Related Work

To make this paper self-contained, we introduce some related topics here on fine-grained entity typing, and copy mechanism.

2.1. Fine-Grained Entity Typing

The FET task was first introduced by [9,28]. Different from lexicon-level [29], discourse-level [30], and corpus-level [31,32] FET, most previous works consider sentence-level entity typing. The progress of FET has primarily focused on the following directions.

Neural Network Model. Different from early heuristic hand-crafted feature-based models [9,28,33] and embedding-based methods [1], the neural models are expected to learn better latent representations for mention and context [8,34,35]. For instance, ref. [36] firstly used recurrent neural networks (RNNs) to recursively obtain a vector representation of each entity mention. Refs. [37,38] proposed to incorporate an attention mechanism with LSTM. Ref. [39] proposed a CNN-based FET model. Refs. [40,41] proposed attentive neural models that also encoded latent type representations besides mention and context. Refs. [8,41] found that the pretrained language model ELMo [42] performed better than BERT [43] as the input of a neural model. Our model builds upon these progress and takes advantage of the state-of-the-art neural network architecture.

Incorporating Knowledge Graphs. Different from distant supervision that only takes advantage of KGs to build training data, a few researchers focused on incorporating KGs into FET models. Ref. [37] proposed to improve FET with knowledge attention which learns the relational information from KGs. Ref. [34] proposed to enrich the mention features by adding a KG-type representation obtained from KGs. However, they did not directly utilize the entity-typing facts in KGs for improving FET.

Denoise. Most typical FET datasets, such as FIGER (GOLD) [9], BBN [24], and OntoNotes [18,28], are labeled with KGs by distant supervision [10], which inevitably bring noise to training data. Several studies aimed to address these problems by heterogeneous partial label embedding [17,44,45], hierarchy-aware loss normalization [38], language model enhancement [46], filtering function [47], compact latent space clustering [48], virtual adversarial learning [49], attentive graph convolution network [50], and automatic

relabeling [51]. There would be a concern about the noise issue for our model; however, it is not the main focus of this paper. We believe that our model could be further boosted by adding a denoising module, which we reserve for future.

Others. There are some other points concerned in FET. Recently, some researchers have focused on FET in KGs, also known as knowledge graph entity typing (KGET), which is a subtask of knowledge graph completion [52], by using external data outside KGs [53,54], or only with structural relational information in KGs [52,55]. Some have concentrated on encoding the hierarchical characteristics of fine-grained type in their models [7,44] and on zero-shot entity typing [56,57].

2.2. Copy Mechanism

The copy mechanism is widely used in various natural language generation tasks, such as sequence-to-sequence learning [58,59], keyphrase generation [25], abstractive summarization [27], and entity prediction [60]. Ref. [58] proposed the pointer networks, which used attention as a pointer that could select a member of an output sequence directly from the input, which could be seen as a copy model with an attention mechanism. However, it could not make prediction using external lexemes besides the input sequence. Ref. [59] proposed COPYNET to solve this issue in a hybrid end-to-end way, which incorporated the copy mechanism with a generation model which yielded external lexemes that did not appear in the input sequence. Based on [25,59] proposed to incorporate a copy mechanism with a recurrent neural network (RNN)-based generation model for deep keyphrase generation. To enhance the copy mechanism, ref. [61] proposed SeqCopyNet, which not only could copy single words but also copy subsequences from the input text. Ref. [27] proposed a transformer model with copy mechanism for abstractive summarization. Inspired by previous works, we utilize the copy mechanism with the typing characteristics of knowledge graphs. To the best of our knowledge, we are the first to attempt to incorporate the copy mechanism in neural fine-grained entity typing.

3. Methodology

In this section, we introduce the details of the proposed model COPYFET. We first give the notations and then introduce the feature encoder and the model architecture, which includes a copy model and a generation model.

Notations. Given an entity mention m and its context c in a sentence s , and a set of type tags \mathcal{T} , our model aims to predict the probability of each type $t \in \mathcal{T}$ for this mention. We denote $w_1, w_2, \dots, [m_1, m_2, \dots, m_n], \dots, w_L$ as the words in the sentence, where m_1, m_2, \dots, m_n denotes the words in entity mention. L and n denote the number of words in the sentence s and mention m . Boldfaced \mathbf{m} and \mathbf{c} represent the embedding vector of mention m and context c , respectively. Besides containing lots of entity facts, a knowledge graph \mathcal{G} also provides a large amount of existing entity-typing instances, i.e., $\mathcal{G} = \{(e, t) | e \in \mathcal{E}, t \in \hat{\mathcal{T}}\}$. \mathcal{E} and $\hat{\mathcal{T}}$ represent the set of entities and types, respectively. For each entity mention m and its corresponding entity e , we build a bounded subset of \mathcal{T} which is specific to m (namely a type's *copying vocabulary* for m) as \mathcal{T}_m . It consists of all the types that have been labeled as the types in known entity-typing facts with entity e in the KG.

Since the labels in type tag set \mathcal{T} and KG type set $\hat{\mathcal{T}}$ may not be exactly the same, it needs type-mapping processing. The *copying vocabulary* \mathcal{T}_m is an N -dimensional few-hot indicative vector. N stands for the size of \mathcal{T} . The value of types in the *copying vocabulary* are marked 1 while others are set to 0. The fine-grained entity typing task is to infer the type probability distribution in type tags space \mathcal{T} given the mention m and context c , i.e.,

$$\mathbf{p}(t|m, c)=?.$$

Feature Encoder. We concatenated the entity mention representation \mathbf{m} and its context representation \mathbf{c} in the sentence as the feature vector \mathbf{x} , as follows:

$$\mathbf{x} = \begin{bmatrix} \mathbf{m} \\ \mathbf{c} \end{bmatrix} \quad (1)$$

Entity Mention Representation. The entity mention representation \mathbf{m} was simply calculated by averaging the embeddings of all words in entity mention $[m_1, m_2, \dots, m_n]$ [37], as follows:

$$\mathbf{m} = \frac{1}{n} \sum_{i=1}^n \mathbf{m}_i, \quad (2)$$

where n represents the length of the entity mention.

Context Representation. We used a bidirectional LSTM (BiLSTM) to encode context representation. We first utilized a special token to denote the mention, as the token “[M]” in Figure 1. Then, the word embeddings of the modified context words $\{w_1, w_2, \dots, [M], \dots, w_L\}$ were fed into two layers of BiLSTM (bidirectional long short-term memory), and the context representation \mathbf{c} was the sum of the BiLSTM layers’ outputs:

$$\mathbf{c} = \mathbf{h}_m^1 + \mathbf{h}_m^2, \quad (3)$$

where \mathbf{h}_m^1 and \mathbf{h}_m^2 are the output of the first and the second layer of BiLSTM for [M], respectively.

3.1. Copy Model

The copy model was designed to identify types from its corresponding *copying vocabulary* that stemmed from the known entity-typing instances in existing KGs.

We first dealt with the training dataset to build the type’s *copying vocabulary* for each mention, i.e., \mathcal{T}_m , which contained three steps: (1) Given a mention m , we utilized a simple entity-linking (EL) algorithm (similar to [34]) to retrieve its corresponding entity in the KG. Specifically, we directly linked the mention to the entity with the largest commonness score [62], which indicated the probability of an entity given the entity mention. The commonness score was calculated based on the anchor links in Wikipedia. (2) If the EL algorithm returned an entity, we obtained the types of this entity from the KG. (3) Since the types in the KG may be different from the target type set \mathcal{T} , following the rules used in [56], we mapped them to the type tags in \mathcal{T} . \mathcal{T}_m was an N-dimensional multi-hot vector and the value of types in the type’s *copying vocabulary* was 1, while others were 0. Note that if the result of EL is NULL, we simply set \mathcal{T}_m as a zero vector.

If the mention m has the type’s *copying vocabulary* \mathcal{T}_m , COPYFET increases the probability value calculated for the candidate types that are chosen from the *copying vocabulary*. Specifically, the copy model first builds an indicator vector $\mathbf{v}_{\mathcal{T}}$ with a multilayer perceptron (MLP):

$$\mathbf{v}_{\mathcal{T}} = \mathbf{f}_{\mathbf{c}}(\mathbf{x}), \quad (4)$$

where $\mathbf{f}_{\mathbf{c}}$ is a three-layer MLP with Relu activation, and the vector $\mathbf{v}_{\mathcal{T}}$ is an N-dimensional indicator vector. N is the size of the type’s tag vocabulary \mathcal{T} .

To decrease the probability value of some types that do not belong to the type’s *copying vocabulary* (i.e., uninterested types for the copy model), COPYFET implements an element-wise multiplication between the index vector $\mathbf{v}_{\mathcal{T}}$ and the indicator vector \mathcal{T}_m . Formally, the type prediction distribution of the copy model is defined as follows:

$$\mathbf{p}_c(t|m, c) = \mathbf{v}_{\mathcal{T}} \odot \mathcal{T}_m, \quad (5)$$

where $\mathbf{p}_c(t|m, c)$ stands for the prediction probabilities on the type’s *copying vocabulary*. The maximum item of $\mathbf{p}_c(t|m, c)$ indicates the type will be copied from the type’s *copying vocabulary*. The basic idea behind the copy model is that it is more beneficial to learn to predict from a small candidate set than the whole type vocabulary. However, entity-

typing facts may be out-of-KG. Thus, it needs an additional generation model to infer such typing facts.

3.2. Generation Model

With the same mention m and context c , the generation model concentrates on type inference by selecting the type from the whole type set \mathcal{T} . The inference made by the generation model treats the typing fact as a new one without any mentions to the knowledge graph. Similar to the copy method, the generation model also builds a whole type vocabulary query vector \mathbf{p}_g as follows:

$$\mathbf{p}_g(t|m, c) = f_g(\mathbf{x}), \quad (6)$$

where f_g is a three-layer MLP with Relu activation. Similar to $\mathbf{p}_c(t|m, c)$ in the copy model, $\mathbf{p}_g(t|m, c)$ stands for the predicted probability distribution among the whole type set. The largest score in $\mathbf{p}_g(t|m, c)$ denotes the corresponding type we inferred in the whole type set from the generation model.

3.3. Incorporating Copy Model with Generation Model for FET

To make type distribution prediction regarding a query $\mathbf{p}(t|m, c) = ?$, both copy model and generation model predict a type among their candidate type sets. As shown in Figure 1, COPYFET incorporates the predicted results from both models as follows:

$$\mathbf{p}(t|m, c) = \lambda * \mathbf{p}_c(t|m, c) + (1 - \lambda) * \mathbf{p}_g(t|m, c), \quad (7)$$

where $\mathbf{p}_c(t|\cdot)$ stands for the copy model, and $\mathbf{p}_g(t|\cdot)$ stands for the generation model. λ is a hyperparameter for the trade-off between copy model and generation model.

To cope with the *overly specific* issue that usually biases the model towards popular subtypes instead of generic ones, i.e., preferring *politician* over *person*, we designed a recursive selecting method. For each stage, we chose the type that had the maximum combined probability in that depth and then we went deeper. We implemented it recursively until the maximum probability was smaller than a threshold. For the sake of simplicity, let us suppose the maximum depth of the type hierarchy is 2 without loss of generality. Formally, we denote \mathcal{T}^1 as the type set of first level, and $\mathcal{T}_{t_i}^2$ stands for the children type set (second level) of a specific type $t_i^1 \in \mathcal{T}^1$. The final output fine-grained entity type prediction \hat{t} changes with a threshold β as follows:

$$\begin{aligned} t_i^1 &= \arg \max_{t \in \mathcal{T}^1} \mathbf{p}(t|m, c), \\ t_j^2 &= \arg \max_{t \in \mathcal{T}_{t_i}^2} \mathbf{p}(t|m, c), \\ \hat{t} &= \mathbf{p}(t = t_j^2|m, c) > \beta ? t_j^2 : t_i^1, \end{aligned} \quad (8)$$

where t_i^1 indicates the output type with the highest probability in the first level, and t_j^2 indicates its subtype with the highest probability in $\mathcal{T}_{t_i}^2$. Here, $\beta \in (0, 1)$ is a hyperparameter acting as a threshold that controls the specific degree of the hierarchical type. The higher β is, the more coarse-grained the final output type should be.

4. Cross-Entropy Loss Function for Optimization

Since in the training set, there were multiple ground truths for one mention, the objective function was defined as the element-wise cross-entropy over all entity mentions:

$$\mathcal{L} = - \sum_i t_i^* \log \mathbf{p}(t) + (1 - t_i^*) \log(1 - \mathbf{p}(t)), \quad (9)$$

where t^* indicates the ground truth types of the mention. Since the training data were automatically generated by linking the mention to all labels in the KG, which was the same as the copy vocabulary, it may cause the model to overfit the weakly labeled training data. This is fine for most types of entities such as */locations* and */organizations* since they usually have the same type in different context; however, this is problematic for other context-dependent entities, such as */person* entity mentions. To alleviate overfitting issue, we added a random fine-grained type label that did not belong to this entity when building the type copying vocabulary. During training, we employed dropout in two LSTM layers and MLP layers.

5. Experiments

In this section, we evaluate the effectiveness of the proposed COPYFET with two public datasets.

5.1. Datasets

We used two publicly available benchmark datasets for FET experiments, including Wiki/FIGER (GOLD) [9] and BBN [24]. The statistics of the percentage of the entity mentions' typing facts of the two benchmark datasets (i.e., testing data) that have been already covered by existing KGs are shown in Table 1, and other statistics are included in Table 2.

Table 2. The statistics of the two benchmark datasets.

Dataset	# Train	# Dev	# Test	# Label	Depth
FIGER [9]	4,932,761	2000	563	128	2
BBN [24]	4,695,789	2000	13,282	56	2

Wiki/FIGER (GOLD). Ref. [9] extracted a dataset from Wikipedia articles and news reports to form the training, validation set, and testing data, and annotated entity mentions using 113 types with a two-level hierarchy.

BBN [24] is based on a portion of the one million word Penn Treebank corpus from Wall Street Journal articles and is completely manually annotated using 56 types with a two-level hierarchy. Ref. [44] regenerated the training corpus via distant supervision.

5.2. Baselines

To demonstrate the effectiveness of our proposed model COPYFET, we compared results with several state-of-the-art FET models:

- AFET [44]: one of the most widely used FET model. AFET models the samples with only one label and samples with multiple labels separately with a partial label loss to handle noisy labels.
- Attentive [63]: a popular attention-based neural network model which uses attention mechanism to focus on relevant information.
- AAA [45]: an extension of AFET which jointly encodes entity mentions and their context representation.
- NFETC [38]: a very popular model which formulates FET as a single-label classification problem with hierarchy-aware loss.
- NFETC-CLSC [48]: an influential extension of NFETC which utilizes imperfect annotation as model regularization via compact latent space clustering to address the confirmation bias problem.
- IFETET [34]: a FET model which utilizes entity type information from a KB obtained through entity linking to form the final feature vector of a mention.
- NDP [7]: a random-walk-based model which weighs out noise with a loss function.
- HFET [41]: a popular ELMo-based pretrained language model which adopts a hybrid type classifier.

- HET [8]: a recent model that takes the hierarchical ontology into account with a multilevel learning-to-rank loss and gains great performance improvement.
- FGET-RR [50]: a recent model that refines the noisy mention representations by attending to corpus-level contextual clues prior to the end classification.
- Box [64]: a recent box-based model for fine-grained entity typing.

5.3. Experimental Settings

In COPYFET, we evaluated the performance by the strict accuracy (Strict Acc), loose macro F-score (Macro-F1), and loose micro F-score (Micro-F1), which are the most widely used evaluation settings for FET systems [9]. We used pretrained word embeddings from [65]. The settings are shown in Table 3. For training our model, we selected the parameters λ and $\beta \in \{0.1, 0.2, 0.5, 0.7\}$ based on the validation set. Finally, the λ was set to 0.5 and 0.7 on Wiki/FIGER (GOLD) and BBN, respectively. The β was set to 0.5 on both Wiki/FIGER (GOLD) and BBN.

Table 3. Hyperparameter settings for training on two datasets.

Parameter	Wiki/FIGER (GOLD)	BBN
Learning rate	1×10^{-3}	1×10^{-3}
Batch size	256	256
Word vector size	300	300
LSTM hidden dropout	0.5	0.5
λ	0.5	0.7
β	0.5	0.5

5.4. Results and Analysis

Table 4 demonstrates the results of fine-grained entity typing. We can observe that our COPYFET outperforms all baselines for fine-grained entity typing in terms of all metrics on Wiki/FIGER and BBN. Specifically, our model improves the strict accuracy on the two datasets with values of 76.4 and 83.6, respectively, which confirms the capability of COPYFET to incorporate the copy mechanism for FET using the copy-generation mechanism and to infer types for mentions in text. These results are in line with our intuition, which indicated that the proposed model was capable of leveraging entity-typing information from existing KGs to build the type's *copying vocabulary*. It is the main feature that leads to the better performance of CopyFet. It can substantially narrow down the optimal list of possible type labels for unlabeled mentions, and thus can significantly improve the performance of FET.

Table 4. Fine-grained entity typing results. Evaluation of different models on Wiki/FIGER and BBN. The results of baselines are directly taken from the original papers. The best scores are in **bold**.

Model	Wiki/FIGER (GOLD)			BBN		
	Strict Acc.	Macro F1	Micro F1	Strict Acc.	Macro F1	Micro F1
AFET [44]	53.3	69.3	66.4	67.0	72.7	73.5
Attentive [63]	59.7	80.0	75.4	48.4	73.2	72.4
AAA [45]	65.8	81.2	77.4	73.3	79.1	79.2
NFETC [38]	68.9	81.9	79.0	72.1	77.1	77.5
NFETC-CLSC [48]	-	-	-	74.7	80.7	80.5
IFETET [34]	74.9	86.2	84.0	82.1	88.1	89.3
NDP [7]	67.7	81.8	78.0	72.7	76.4	77.7
HFET [41]	62.9	83.0	79.8	55.9	79.3	78.1
HET [8]	65.5	80.5	78.1	75.2	79.7	80.5
FGET-RR [50]	71.0	84.7	80.5	70.3	81.9	82.3
Box [64]	-	79.4	75.0	-	78.7	78.0
COPYFET (Ours)	76.4	86.7	84.6	83.6	89.4	89.9

5.5. Ablation Study

To evaluate the different component of our model COPYFET, we conducted an ablation study. To this end, we generated a variant of COPYFET by deleting the use of its copy components (called COPYFET-Generation-only), and compared the fine-grained entity-typing performance on Wiki/FIGER and BBN. Table 5 shows the fine-grained entity-typing results by the variant of our model COPYFET. We can observe that the copy model is significant. Deleting the copy module leads to drops of all metrics on strict accuracy in Wiki/FIGER and BBN, respectively, which indicates that learning to infer types for unlabeled mentions by referring to the known entity-typing facts in KGs can be quite beneficial. The results demonstrate our model COPYFET can successfully take into account the known entity-typing information from existing KGs via the copy mechanism to improve FET. Next, we analyzed the detailed results of COPYFET compared to COPYFET-Generation-only. In Figure 2, we present the type-wise performance for the top-10 most frequent types in the FIGER testing dataset. Compared to COPYFET-Generation-only, COPYFET performs better in all types.

Table 5. Ablation study. Fine-grained entity typing results by different variants of our model COPYFET on Wiki/FIGER and BBN.

Model	Wiki/FIGER (GOLD)			BBN		
	Strict Acc.	Macro F1	Micro F1	Strict Acc.	Macro F1	Micro F1
COPYFET-Generation-only	69.9	82.7	80.6	79.8	86.8	87.9
COPYFET	76.4	86.7	84.6	83.6	89.4	89.9

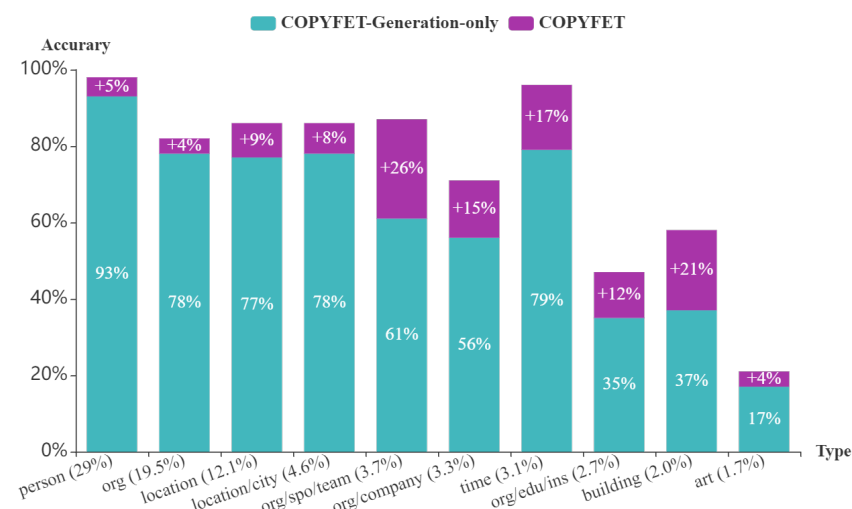


Figure 2. Performance analysis of COPYFET on the top 10 types present in FIGER dataset. In these ten types, COPYFET performs better than COPYFET-Generation-only.

5.6. Case Study

Table 6 gives two examples of fine-grained entity-typing results on the FIGER and BBN testing set. For example, given a mention of “Unitec Institute of Technology” and its context, the possible fine-grained types are inferred by COPYFET-Generation-only and COPYFET, respectively. The former makes a false prediction (i.e., /location), while the latter can make a positive inference since the known entity-typing fact (“UNITEC”, /organization/educational_institution) has already been included in the KG (e.g., Freebase). This is quite helpful for the prediction, which illustrates the efficacy of the copy model in COPYFET.

Table 6. Example type predictions on FIGER and BBN testing sets using COPYFET-Generation-only and COPYFET. **Bold** indicates the true prediction.

Data	Mention and Context	Known Facts in KGs	COPYFET-Generation-only	COPYFET
Wiki	The study is from the <i>Unitec Institute of Technology</i> , Auckland, New Zealand.	(UNITEC, /organization) (UNITEC, /organ./edu-cational_inst.)	/location	organization/educational_institution
BBN	The Fleet Street reaction was captured in the Guardian headline, “ <i>Departure Reveals Thatcher Poison.</i> ”	(D. R. T. P., /art) (D. R. T. P., /work_of_art)	/organization	/work_of_art

6. Conclusions and Future Work

In this paper, we proposed a novel model architecture for fine-grained entity typing using KGs. The proposed model leveraged the popular copying mechanism that “copies” an inferred fine-grained type probability distribution of the target from a knowledge graph. The copied distribution was then combined with the output of a regular generation model that predicted the distribution of the full set of types. This new method achieved new SOTA results on FET, outperforming previous methods based on weak supervision or knowledge injection using KGs. The ablation analysis showed that the copying module of the model contributed significantly to the improved prediction quality. Interesting future work directions include exploring how to make the construct operation of mapping from the target types to the KG types in this model soft (currently it is one-hot) and trainable and adding a denoising module in the copy-generation networks.

Author Contributions: Investigation, A.Z. and H.F.; methodology, Z.Y. and Y.Z.; software, A.Z. and H.D.; supervision, Y.Z.; writing—original draft, Z.Y.; writing—review and editing, Z.Y. and S.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China under grant no. 61906159, and in part supported by the Fundamental Research Funds for the Central Universities under grant no. kjc20210102.

Acknowledgments: The authors would like to thank all anonymous reviewers for their comments.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yogatama, D.; Gillick, D.; Lazic, N. 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, Beijing, China, 26–31 July 2015; pp. 291–296.
2. Corro, D.; Luciano, A.; Abujabal, A.; Gemulla, R.; Weikum, G. Finet: Context-aware fine-grained named entity typing. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 868–878.
3. Collins, M.; Singer, Y. Unsupervised models for named entity classification. In Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, College Park, MD, USA, 21–22 June 1999.
4. Ratnikov, L.; Roth, D. Design challenges and misconceptions in named entity recognition. In Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009), Boulder, CO, USA, 4–5 June 2009; pp. 147–155.
5. Choi, E.; Levy, O.; Choi, Y.; Zettlemoyer, L.S. Ultra-fine entity typing. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Melbourne, Australia, 15–20 July 2018.
6. Hou, F.; Wang, R.; Zhou, Y. Transfer learning for fine-grained entity typing. *Knowledge Inf. Syst.* **2021**, *63*, 845–866. [[CrossRef](#)]
7. Wu, J.; Zhang, R.; Mao, Y.; Guo, H.; Huai, J. Modeling Noisy Hierarchical Types in Fine-Grained Entity Typing: A Content-Based Weighting Approach. In Proceedings of the International Joint Conferences on Artificial Intelligence Organization, Beijing, China, 10–16 August 2019; pp. 5264–5270.
8. Chen, T.; Chen, Y.; Durme, B.V. Hierarchical Entity Typing via Multi-level Learning to Rank. In Proceedings of the Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 8465–8475.
9. Ling, X.; Weld, D.S. Fine-grained entity recognition. In Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, Toronto, ON, Canada, 22–26 July 2012.

10. Mintz, M.; Bills, S.; Snow, R.; Jurafsky, D. 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 2009, Singapore, 2–7 August 2009; pp. 1003–1011.
11. Liu, Y.; Liu, K.; Xu, L.; Zhao, J. Exploring fine-grained entity type constraints for distantly supervised relation extraction. In Proceedings of the InProceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers 2014, Dublin, Ireland, 11 August 2014; pp. 2107–2116.
12. Toral, A.; Noguera, E.; Llopis, F.; Munoz, R. Improving question answering using named entity recognition. In Proceedings of the International Conference on Application of Natural Language to Information Systems, Alicante, Spain, 15–17 June 2005; pp. 181–191.
13. Yahya, M.; Berberich, K.; Elbassuoni, S.; Weikum, G. Robust question answering over the web of linked data. In Proceedings of the 22nd ACM International Conference on Information & Knowledge Management 2013, San Francisco, CA, USA, 27 October–1 November 2013; pp. 1107–1116.
14. Dong, X.; Gabrilovich, E.; Heitz, G.; Horn, W.; Lao, N.; Murphy, K.; Strohmann, T.; Sun, S.; Zhang, W. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 2014, New York, NY, USA, 24–27 August 2014; pp. 601–610.
15. Li, D.; Liu, H.; Zhang, Z.; Lin, K.; Fang, S.; Li, Z.; Xiong, N.N. CARM: Confidence-aware recommender model via review representation learning and historical rating behavior in the online platforms. *Neurocomputing* **2021**, *455*, 283–296. [\[CrossRef\]](#)
16. Liu, H.; Zheng, C.; Li, D.; Shen, X.; Lin, K.; Wang, J.; Zhang, Z.; Zhang, Z.; Xiong, N.N. EDMF: Efficient Deep Matrix Factorization with Review Feature Learning for Industrial Recommender System. *IEEE Trans. Ind. Inform.* **2021**, *18*, 4361–4371. [\[CrossRef\]](#)
17. Ren, X.; He, W.; Qu, M.; Voss, C.R.; Ji, H.; Han, J. 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 2016, San Francisco, CA, USA, 13–17 August 2016; pp. 1825–1834.
18. Weischedel, R.; Hovy, E.; Marcus, M.; Palmer, M.; Belvin, R.; Pradhan, S.; Ramshaw, L.; Xue, N. Ontonotes: A large training corpus for enhanced processing. In *Handbook of Natural Language Processing and Machine Translation: DARPA Global Autonomous Language Exploitation*; Springer: Berlin/Heidelberg, Germany, 2011.
19. Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; Taylor, J. Freebase: A collaboratively created graph database for structuring human knowledge. In Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data 2008, Vancouver, BC, Canada, 10–12 June 2008; pp. 1247–1250.
20. Suchanek, F.M.; Kasneci, G.; Weikum, G. Yago: A core of semantic knowledge. In Proceedings of the 16th International Conference on World Wide Web 2007, Banff, AB, Canada, 8–12 May 2007; pp. 697–706.
21. Lehmann, J.; Isele, R.; Jakob, M.; Jentzsch, A.; Kontokostas, D.; Mendes, P.N.; Hellmann, S.; Morse, M.; van Kleef, P.; Auer, S.; et al. DBpedia: A largescale, multilingual knowledge base extracted from Wikipedia. *Semant. Web J.* **2015**, *6*, 167–195. [\[CrossRef\]](#)
22. Weischedel, R.; Palmer, M.; Marcus, M.; Hovy, E.; Pradhan, S.; Ramshaw, L.; Xue, N.; Taylor, A.; Kaufman, J.; Franchini, M.; et al. OntoNotes Release 5.0. Abacus Data Network. 2013.
23. Ding, N.; Xu, G.; Chen, Y.; Wang, X.; Han, X.; Xie, P.; Zheng, H.T.; Liu, Z. Few-nerd: A few-shot named entity recognition dataset. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Online, 1–6 August 2021; pp. 3198–3213.
24. Weischedel, R. *BBN Pronoun Coreference and Entity Type Corpus*; Linguistic Data Consortium: Philadelphia, PA, USA, 2005; p. 112.
25. Meng, R.; Zhao, S.; Han, S.; He, D.; Brusilovsky, P.; Chi, Y. Deep keyphrase generation. In Proceedings of the Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vancouver, BC, Canada, 30 July–4 August 2017; pp. 582–592.
26. See, A.; Liu, P.J.; Manning, C.D. Get to the point: Summarization with pointer-generator networks. In Proceedings of the Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vancouver, BC, Canada, 30 July–4 August 2017.
27. Xu, S.; Li, H.; Yuan, P.; Wu, Y.; He, X.; Zhou, B. Self-Attention Guided Copy Mechanism for Abstractive Summarization. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 1355–1362.
28. Gillick, D.; Lazić, N.; Ganchev, K.; Kirchner, J.; Huynh, D. Context-dependent fine-grained entity type tagging. *arXiv* **2014**, arXiv:1412.1820.
29. Yao, L.; Riedel, S.; McCallum, A. Universal schema for entity type prediction. In Proceedings of the 2013 Workshop on Automated Knowledge Base Construction, San Francisco, CA, USA, 27–28 October 2013; pp. 79–84.
30. Zhang, S.; Duh, K.; Durme, B.V. Fine-grained Entity Typing through Increased Discourse Context and Adaptive Classification Thresholds. In Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics, New Orleans, LA, USA, 5–6 June 2018; pp. 173–179.
31. Yaghoobzadeh, Y.; Schütze, H. Corpus-level Fine-grained Entity Typing Using Contextual Information. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP2015), Lisbon, Portugal, 17–21 September 2015; pp. 715–725.
32. Yaghoobzadeh, Y.; Adel, H.; Schütze, H. Corpus-level fine-grained entity typing. *J. Artif. Intell. Res.* **2018**, *61*, 835–862. [\[CrossRef\]](#)
33. Yosef, M.A.; Bauer, S.; Hoffart, J.; Spaniol, M.; Weikum, G. Hyena: Hierarchical type classification for entity names. In Proceedings of the COLING 2012: Posters, Mumbai, India, 8–15 December 2012; pp. 1361–1370.

34. Dai, H.; Du, D.; Li, X.; Song, Y. Improving Fine-grained Entity Typing with Entity Linking. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; pp. 6210–6215.
35. Xiong, W.; Wu, J.; Lei, D.; Yu, M.; Chang, S.; Guo, X.; Wang, W.Y. Imposing Label-Relational Inductive Bias for Extremely Fine-Grained Entity Typing. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, MN, USA, 2–7 June 2019; pp. 773–784.
36. Dong, L.; Wei, F.; Sun, H.; Zhou, M.; Xu, K. A hybrid neural model for type classification of entity mentions. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, MN, USA, 2–7 June 2019; pp. 1243–1249.
37. Xin, J.; Lin, Y.; Liu, Z.; Sun, M. Improving Neural Fine-Grained Entity Typing with Knowledge Attention. In Proceedings of the AAAI Conference on Artificial Intelligence 2018, New Orleans, LA, USA, 2–7 February 2018; pp. 5997–6004.
38. Xu, P.; Barbosa, D. Neural fine-grained entity type classification with hierarchy-aware loss. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), New Orleans, LA, USA, 1–6 June 2018; pp. 16–25.
39. Murty, S.; Verga, P.; Vilnis, L.; Radovanovic, I.; McCallum, A. Hierarchical Losses and New Resources for Fine-grained Entity Typing and Linking. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) 2018, Melbourne, Australia, 15–20 July 2018; pp. 97–109.
40. Yuan, Z.; Downey, D. OType: A neural architecture for open named entity typing. In Proceedings of the AAAI Conference on Artificial Intelligence 2018, New Orleans, LA, USA, 2–7 February 2018.
41. Lin, Y.; Ji, H. An Attentive Fine-Grained Entity Typing Model with Latent Type Representation. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; pp. 6197–6202.
42. Peters, M.E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep contextualized word representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), New Orleans, LA, USA, 1–6 June 2018; pp. 2227–2237.
43. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186.
44. Ren, X.; He, W.; Qu, M.; Huang, L.; Ji, H.; Han, J. AFET: Automatic fine-grained entity typing by hierarchical partial-label embedding. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing 2016, Austin, TX, USA, 1–4 November 2016.
45. Abhishek, A.; Anand, A.; Awekar, A. Fine-grained entity type classification by jointly learning representations and label embeddings. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics 2017, Valencia, Spain, 3–7 April 2017; pp. 797–807.
46. Xin, J.; Zhu, H.; Han, X.; Liu, Z.; Sun, M. Put It Back: Entity Typing with Language Model Enhancement. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 993–998.
47. Onoe, Y.; Durrett, G. Learning to Denoise Distantly-Labeled Data for Entity Typing. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, MN, USA, 2–7 June 2019; pp. 2407–2417.
48. Chen, B.; Gu, X.; Hu, Y.; Tang, S.; Hu, G.; Zhuang, Y.; Ren, X. Improving Distantly-supervised Entity Typing with Compact Latent Space Clustering. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, MN, USA, 2–7 June 2019; pp. 2862–2872.
49. Shi, H.; Tang, S.; Gu, X.; Chen, B.; Chen, Z.; Shao, J.; Ren, X. Alleviate Dataset Shift Problem in Fine-grained Entity Typing with Virtual Adversarial Training. In Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence 2021, Yokohama, Japan, 11–17 July 2020.
50. Ali, M.A.; Sun, Y.; Li, B.; Wang, W. Fine-Grained Named Entity Typing over Distantly Supervised Data Based on Refined Representations. In Proceedings of the AAAI Conference on Artificial Intelligence 2020, New York, NY, USA, 7–12 February 2020; pp. 7391–7398.
51. Zhang, H.; Long, D.; Xu, G.; Zhu, M.; Xie, P.; Huang, F.; Wang, J. Learning with Noise: Improving Distantly-Supervised Fine-grained Entity Typing via Automatic Relabeling. In Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence 2021, Yokohama, Japan, 11–17 July 2020.
52. Zhao, Y.; Zhang, A.; Xie, R.; Liu, K.; Wang, X. Connecting Embeddings for Knowledge Graph Entity Typing. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 6419–6428.
53. Yaghoobzadeh, Y.; Schütze, H. Multi-Multi-View Learning: Multilingual and Multi-Representation Entity Typing. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, online, 5–10 July 2020; pp. 3060–3066.

54. Jin, H.; Hou, L.; Li, J.; Dong, T. Fine-Grained Entity Typing via Hierarchical Multi Graph Convolutional Networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; pp. 4969–4978.
55. Moon, C.; Jones, P.; Samatova, N.F. Learning Entity Type Embeddings for Knowledge Graph Completion. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, Singapore, 6–10 November 2017; pp. 2215–2218.
56. Zhou, B.; Khashabi, D.; Tsai, C.T.; Roth, D. Zero-Shot Open Entity Typing as Type-Compatible Grounding. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2020; pp. 2065–2076.
57. Obeidat, R.; Fern, X.; Shahbazi, H.; Tadeipalli, P. Description-Based Zero-shot Fine-Grained Entity Typing. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, MN, USA, 2–7 June 2019; pp. 807–814.
58. Vinyals, O.; Fortunato, M.; Jaitly, N. Pointer networks. In Proceedings of the Advances in Neural Information Processing Systems 28 (NIPS 2015), Montreal, QC, Montreal, 7–12 December 2015; pp. 2692–2700.
59. Gu, J.; Lu, Z.; Li, H.; Li, V.O.K. Incorporating copying mechanism in sequence-to-sequence learning. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Berlin, Germany, 7–12 August 2016; pp. 1631–1640.
60. Zhu, C.; Chen, M.; Fan, C.; Cheng, G.; Zhang, Y. Learning from History: Modeling Temporal Knowledge Graphs with Sequential Copy-Generation Networks. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtually, 2–9 February 2021.
61. Zhou, Q.; Yang, N.; Wei, F.; Zhou, M. Sequential copying networks. In Proceedings of the AAAI Conference on Artificial Intelligence 2018, New Orleans, LA, USA, 2–7 February 2018.
62. Pan, X.; Cassidy, T.; Hermjakob, U.; Ji, H.; Knight, K. Unsupervised entity linking with abstract meaning representation. In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies 2015, Denver, CO, USA, 31 May–5 June 2015; pp. 1130–1139.
63. Shimaoka, S.; Stenetorp, P.; Inui, K.; Riedel, S. An attentive neural architecture for fine-grained entity type classification. In Proceedings of the 5th Workshop on AKBC, San Diego, CA, USA, 17 June 2016; pp. 69–74.
64. Onoe, Y.; Boratko, M.; McCallum, A.; Durrett, G. Modeling Fine-Grained Entity Types with Box Embeddings. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Online, 1–6 August 2021; pp. 2051–2064.
65. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.