A Neighborhood-Attention Fine-grained Entity Typing for Knowledge Graph Completion

Jianhuan Zhuo
zhuojianhuan@iie.ac.cn
Institute of Information Engineering, Chinese Academy of Sciences,Beijing,China & School of Cyber Security, University of Chinese Academy of Sciences,Beijing,China

Qiannan Zhu*
zhuqiannan@ruc.edu.cn
Gaoling School of Artificial Intelligence, Renmin University of China, Beijing Key Laboratory of Big Data Management and Analysis Methods

Yinliang Yue*
yueyinliang@iie.ac.cn
Institute of Information Engineering, Chinese Academy of Sciences,Beijing,China & School of Cyber Security, University of Chinese Academy of Sciences,Beijing,China

Yuhong Zhao
Weisi Han
zhaoyuhong@iie.ac.cn
hanweisi@iie.ac.cn
Institute of Information Engineering, Chinese Academy of Sciences,Beijing,China

ABSTRACT

Knowledge graph (KG) entity typing focuses on inferring possible entity type instances, which is a significant subtask of knowledge graph completion (KGC). Existing entity typing methods usually exploit the entity representation to model the transmission between entities and their types, which cannot fully explore the fine-grained entity typing on identifying the semantic type of an entity. To address these issues, we propose Neighborhood-Attention Neural Fine-Grained Entity Typing (AttEt), which considers the neighborhood information of the entities from KGs to bridge entities and their types together. In this paper, AttEt first develops a type-specific attention mechanism to aggregate the neighborhood knowledge of the given entity with type-specific weights. These weights are beneficial to capture various characteristics for different types of the entity, and further imply the complex correlation among these fine-grained types. Then, AttEt adaptively integrates the aggregated neighbor-level representation with entity inherent embedding to calculate the matching score between the entity and its candidate type. Besides, many entities are sparse in their relations with other entities in KGs, which makes the entity typing task more challenging. To solve this problem, we present a smooth strategy on relation-sparsity entities to improve the robustness of the model. Extensive experiments on two real-world datasets (Freebase and YAGO) show that AttEt significantly outperforms state-of-the-art baselines in the HITS@1 by 2.11% on Freebase and by 8.42% on YAGO, respectively.

*Corresponding author

WSDM '22, February 21–25, 2022, Tempe, AZ, USA.
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9132-0/22/02…$15.00
https://doi.org/10.1145/3488560.3498395

CCS CONCEPTS

• **Computing methodologies** → **Knowledge representation and reasoning**.

KEYWORDS

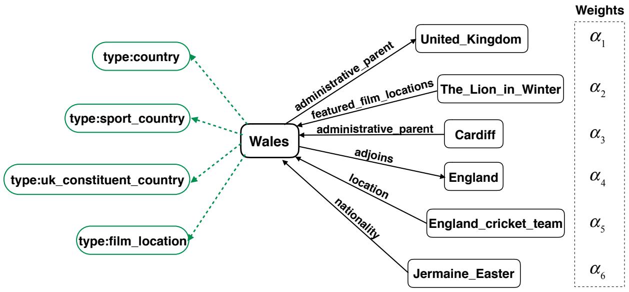
Knowledge Graph Completion, Fine-grained Entity Typing, Knowledge Graph Representation Learning, Entity Type Prediction

ACM Reference Format:
Jianhuan Zhuo, Qiannan Zhu, Yinliang Yue, Yuhong Zhao, and Weisi Han. 2022. A Neighborhood-Attention Fine-grained Entity Typing for Knowledge Graph Completion. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining (WSDM '22), February 21–25, 2022, Tempe, AZ, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3488560.3498395

1 INTRODUCTION

Knowledge graphs (KGs) like DBpedia [15], YAGO [29] and Freebase [1], have gradually played a critical role in many knowledge-driven applications, such as question answering [6, 9], recommendation systems [28], dialogue generation [18], etc. Unfortunately, KGs usually suffer from incompleteness because they are typically populated automatically by mining from semi-structured or unstructured data sources. For such incompleteness issue, a lot of works [3, 16, 21, 45] have been proposed to complete knowledge graphs. Nevertheless, they mainly focus on inferring missing entities or relationships but rarely pay attention to the completion of entity types. In practice, entity types are important and widely used in various applications such as relation extraction [10, 46], coreference resolution [8] and recommendation system [34].

Recently, extensive studies [20, 47] that focus on inferring entity types using KGs have been presented and achieved promising performance. These studies usually learn representations of entities and types by leveraging translation-based methods [3, 17] or GCN-based methods [12, 13], and then build the correlation between the entities and their types. In representation learning stage, these methods model the entity's representation by $\mathbf{e} = f_1(\{\alpha_i, \varphi_i\}), i = 1, \dots n$, where α_i is the importance of the neighborhood φ_i of the entity e and n is the number of neighborhood of entity e. Concretely,



Weights Distribution on Neighborhoods

TYPES	type:country	type:uk_constituent_country
Translation-based Methods	$\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = \alpha_5 = \alpha_6$	$\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = \alpha_5 = \alpha_6$
GCN-based Methods	$\alpha_6 \geq \alpha_4 \geq \alpha_1 \geq \alpha_5 \geq \alpha_2 \geq \alpha_3$	$\alpha_6 \geq \alpha_4 \geq \alpha_1 \geq \alpha_5 \geq \alpha_2 \geq \alpha_3$
Our	$\alpha_3 \geq \alpha_6 \geq \alpha_4 \geq \alpha_1 \geq \alpha_5 \geq \alpha_2$	$\alpha_1 \geq \alpha_4 \geq \alpha_3 \geq \alpha_6 \geq \alpha_5 \geq \alpha_2$

Figure 1: Illustration of the entity *Wales* with its neighborhoods and types. The table on the bottom is the weights distribution of entity on different neighborhoods for two types.

the translation-based methods treat entity’s neighborhoods equally and assign the same weight on the importance distribution, i.e., $\alpha_1 = \alpha_2 = \cdots = \alpha_n$. While GCN-based methods enrich entity’s embedding by fusing all the embeddings of its neighbors with different weights, like $\alpha_6 \geq \alpha_4 \geq \alpha_1 \cdots \geq \alpha_3$. In the inference stage, both methods apply another classification model $f_2(e, t_e)$ to make type inference, where t_e is the candidate type of the entity e.

However, when inferring different types of the same entity, the weight distribution on the neighbors learned from the above methods is undifferentiated, which is not fully to exploit the complex interdependencies between types, such as subtype, intersection and disjunction. As shown in Figure 1, for the entity *Wales*, its four types (*country*, *uk_constituent_country*, *sport_country*, *film_location* [1]) imply the hierarchies of these fine-grained types. For example, the type *uk_constituent_country* is the subtype of the type *country*, and the two types *uk_constituent_country* and *sport_country* are intersected with each other as they are overlapped on some topics. The bottom table gives the weight distribution on neighborhoods of the entity *Wales*. Empirically, for inferring different types like *country* and *uk_constituent_country*, the translation-based and GCN-based methods have the same importance distribution, i.e., $\{\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = \alpha_5 = \alpha_6\}$ and $\{\alpha_6 \geq \alpha_4 \geq \alpha_1 \geq \alpha_5 \geq \alpha_2 \geq \alpha_3\}$, respectively. Factually, the neighborhoods an entity has can provide different importance to infer different fine-grained entity types. For example, the neighborhood (*Wales, administrative_parent, United_Kingdom*) has more important roles in revealing the type *uk_constituent_country* because the *Wales* is a constituent country of the *United_Kingdom*, and the neighborhood (*Cardiff, administrative_parent, Wales*) are more likely to infer the type *country* because the *Cardiff* is the capital city of the *Wales*. That is, the weights of such two neighborhood should be larger than other neighborhoods when predicting types *country* and *uk_constituent_country*, respectively. Thus, each type label of an entity should possess its own importance distribution on neighborhoods, which can well capture the characteristics of fine-grained types and reveal the correlation among them.

In this paper, we focus on fine-grained entity typing (FET) for KG completion. To solve this problem, we propose Neighborhood-Attention Neural Fine-Grained Entity Typing model AttEt for inferring entity types. Specifically, AttEt first designs a learnable type-specific attention mechanism on the neighborhood context of the entity, and learns the neighbor-level representation of the entity by aggregating the neighborhoods with type-specific attention weights. These type-specific attention weights on neighborhoods can imply the characteristics of the fine-grained types. Then, AttEt develops a neural component to obtain the comprehensive representation by fusing the entity’s inherent representation and its neighbor-level representation. Finally, AttEt estimates the matching score between the entity and its candidate type to infer whether the candidate type matches the entity. In addition, to maintain the robustness of our model when facing the relation-sparsity entities that have sparse neighborhoods in KGs, we design a smooth factor to refine the attention scores for entity typing. We also empirically apply AttEt to two real-world datasets Freebase and YAGO with extensive experiments. The results show that AttEt achieves consistent gains over state-of-the-art methods.

2 PROBLEM STATEMENT

This paper focuses on entity typing without using external knowledge for KG completion. Formally, we consider a knowledge graph G containing triplets of the form $(e, r, \bar{e})$ and entity type instances of the form (e, t_e), where $e, \bar{e} \in \mathcal{E}, r \in \mathcal{R}, t_e \in \mathcal{T}$, and $\mathcal{E}, \mathcal{R}, \mathcal{T}$ are the entity set, relation set and entity type set, respectively. Each triplet $(e, r, \bar{e})$ represents that the head entity e and tail entity $\bar{e}$ are connected with the relation r and each entity type instance (e, t_e) represents the entity e has the entity type t_e.

Usually, entities in KGs have multiple types. The neighborhoods of entities can provide more valuable and richer information about which candidate types should be attached to the entity. Therefore, this work aims to infer missing entity type instances under considering the neighborhood context of entities, i.e., the entity typing task is to estimate the match probability between the entity e and its candidate entity type t_e by $s = g(e, t_e | G, \mathcal{N}(e))$, where g is our entity typing framework and $\mathcal{N}(e)$ is the neighborhood context of the entity e.

3 MODEL

Our model first learns the neighbor-level representations of the entities by a type-specific attention mechanism, and the entities’ inherent representations (i.e., entity-level) by the translation-based method [3], respectively. Then, our model integrates the neighbor-level and entity-level representations to estimate the match probability between the entities and their candidate types. Figure 2 illustrates the framework of our model.

[1] The prefix of types are trimmed for well-understood and concise description. For instance, the type *uk_constituent_country* stands for the full name */location/uk_constituent_country* in real dataset.

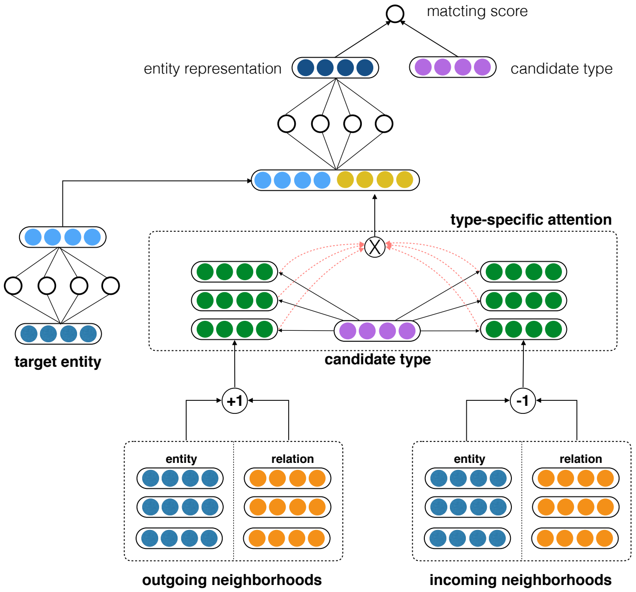


Figure 2: Simple visualization of AttEt architecture.

3.1 Encoding Neighborhoods in KGs

As mentioned above, the neighborhood could make different contributions in inferring different types of the entity. Therefore, we adopt a type-specific attention mechanism to assign type-specific weights on neighborhoods of the entity, and aggregate such neighborhoods with type-specific weights to obtain the neighbor-level representation of the entity afterward.

The relations of a given entity linked with other entities capture its context and provide us with valuable information about what is the type of entity[33]. In this paper, different from inferring types only by using relations in neighborhoods of the target entity, we combine both relations and entities in neighborhoods of the target entity to provide more fine-grained inference. Furthermore, the entities in neighborhoods, especially, are promoted to distinguish hierarchies of the fine-grained entity type labels. Taking the entity *Wales* in Figure 1 as an example, without considering the neighbor entities {*Cardiff, United_Kingdom* } connected with the relation *administrative_parent*, it is inefficient to infer how large the administrative area of *Wales* is, like a country or a city. When considering the neighbor entity *Cardiff* city of the entity *Wales*, it is more likely to assign the larger area type *country* to *Wales*, while prompting to assign a smaller area type *uk_constituent_country* for *Wales* with the neighbor entity *United_Kingdom*.

For brevity, we define the neighborhood set of the entity e_i as $\mathcal{N}(e_i) = \{(r_{i1}, e_{i1}), (r_{i2}, e_{i2}), ..., (r_{in}, e_{in})\}$, where r_{ij} with $j = 1, ..., n$ is the relation that links from entity e_i to e_{ij} or vice versa, and n is the number of neighborhoods of the entity e_i. Inspired by the typical embedding methods [3, 17], we also distinguish the incoming and outgoing [2] neighborhoods of the entities in embedding learning. Thus, we define the following indicator function to strengthen the neighborhood set of entity e_i as

$$f_{ij} = \begin{cases} +1, & (e_i, r_{ij}, e_{ij}) \text{ } is \text{ } hold \\ -1, & (e_{ij}, r_{ij}, e_i) \text{ } is \text{ } hold \end{cases} \tag{1}$$

Having established the neighborhood set with its indicator function, we develop the type-specific attention mechanism to explore the weight distribution of the neighborhoods for predicting fine-grained type labels. First, we use two shared linear transformations, parameterized by two weight matrices $\mathbf{W}_q$ and $\mathbf{W}_k$, to compute the attention coefficients $\alpha_{i,j}$ between the candidate type t_{e_i} of the entity e_i and its neighborhood entity e_{ij} with relation r_{ij}, i.e.,

$$\alpha_{i,j} = \frac{1}{z_i}(\mathbf{W}_q \mathbf{t}_{e_i})^\top \mathbf{W}_k(\mathbf{e}_{ij} + f_{ij} * \mathbf{r}_{ij}) \tag{2}$$

where $\mathbf{t}_{e_i}$ is the type embedding of entity e_i, $\mathbf{W}_q \in \mathbb{R}^{d \times d}$ is the query parameter matrix, $\mathbf{W}_k \in \mathbb{R}^{d \times d}$ is the key parameter matrix and z_i is the normalized factor as

$$z_i = \sum_{(r_{ij}, e_{ij}) \in \mathcal{N}(e_i)} (\mathbf{W}_q \mathbf{t}_{e_i})^\top \mathbf{W}_k(\mathbf{e}_{ij} + f_{ij} * \mathbf{r}_{ij}), j \in [1, ..., n] \tag{3}$$

The type-specific attention mechanism makes different types of the entity have the different weight distribution $\{\alpha_{i,1}, \alpha_{i,2}, ..., \alpha_{i,n}\}$ on its neighborhoods. The type-specific weight distribution naturally implys the correlation among fine-grained type labels (discussed in section 4.3). This is different from previous translation-based and GCN-based methods on using neighborhoods to infer types.

Importantly, there exist many relation-sparsity entities that are sparse in their relations with other entities in KGs, which typically have a small number of neighborhood triples but a large number of types. Their inference lack related semantical support of the neighborhoods. According to the statistics, there are about 4.73% of entities whose types are five times more than their neighborhoods in YAGO43kET. For example, the entity Tina_Turner only has two neighbors (*Tina_Turner*, *hasMusicalRole*, *wordnet_vocal_music_107282006*), (*Tina_Turner*, *hasGender*, *female*) and 70 entity types, such as *wikicat_Swiss_dancers* and *wikicat_20th-century_women_writers*. Obviously, it is hard to support all entity types only by the small scale of neighborhoods of the relation-sparsity entities. Thus, for improving the robustness of the model when facing the relation-sparsity entities, we introduce a smoothing factor λ to refine the attention coefficient as

$$\alpha'_{i,j} = (1 - \lambda)\alpha_{i,j} + \lambda \frac{1}{n} \tag{4}$$

The smooth factor is a balance between the weights from the type-specific attention mechanism and the weights of the averaged number of neighbors. Specifically, the model is beneficial to the relation-dense entities when $\lambda = 0$, and is helpful to the relation-sparsity entities when $\lambda = 1$. Intuitively, there are different optimal λ for datasets with different sparsity. The case study in section 4.4 proves the above intuition. Besides, to constrain the model complexity and avoid overfitting, we employ a bias neighborhood (r_b, e_b) for every entity $e_i \in \mathcal{E}$ in practice. Finally, the neighbor-level representation of the target entity e_i is computed by a linear combination of its neighborhoods with their type-specific weight coefficients $\alpha'_{i,j}$, i.e.,

$$\mathbf{H}^{neighbor} = \sum_{(r_{ij}, e_{ij}) \in \mathcal{N}(e_i)} \alpha'_{i,j} \mathbf{W}_v(\mathbf{e}_{ij} + f_{ij} * \mathbf{r}_{ij}) \tag{5}$$

[2] The outgoing neighborhoods are the neighbor entities with relations linked from e_i to the neighbor entities, and incoming neighborhoods have the relations in opposite direction.

where $\mathbf{W}_v \in \mathbb{R}^{d\times d}$ is the value parameter matrix of type-specific attention module.

3.2 Encoding Triplets in KGs

As studied in [47], the raw embeddings of entities learned from translation-based model [3, 35] imply the KGs' inherent properties, which are also helpful for encoding entity type assertions. Thus, we employ a non-linear transformation to extract the entity-level semantic representation from the raw embeddings of entities trained by TransE [3], i.e.,

$$\mathbf{H} = \sigma(\mathbf{W}\mathbf{e}_i + \mathbf{b}) \tag{6}$$

where $\mathbf{W} \in \mathbb{R}^{d\times d}, \mathbf{b} \in \mathbb{R}^{d}$ are the weight parameters, and σ is the activate function, which can be sigmoid, relu and tanh, etc.

Significantly, fusing a strong embedding model like TransX [11, 36] and DKRL [37] to gain improvement is not our core contribution. For the fair comparison with baselines, we just choose TransE [3] as the basic embedding approach in our experiments.

3.3 Entity Type Prediction

In order to fully explore the information of entities and their neighborhoods, we first incorporate the neighbor-level and entity-level representation of the entity e_i into a neural network to get the final entity representation as

$$\hat{\mathbf{t}}_{e_i} = \mathrm{MLP}([\mathbf{H}; \mathbf{H}^{neighbor}]) \tag{7}$$

Furthermore, we define the prediction function to measure the matching score between entity e_i and its candidate entity type t_{e_i} as

$$s(e_i, t_{e_i}) = \|\hat{\mathbf{t}}_{e_i} - \mathbf{t}_{e_i}\|_2 \tag{8}$$

The prediction function uses the L_2-norm that expects lower scores for the positive samples and higher scores for the negative samples.

3.4 Optimization

To distinguish positive and negative samples, our model is driven to optimize the margin-based ranking loss to lower the scores of positive instances than those of negative ones. In equation (9), the loss function takes a pair of the positive and negative samples as input each time and expects a score gap between them.

$$\mathcal{L} = \sum_{(e_i,t_{e_i})\in\Delta_{e_i}} \sum_{(e_i,t'_{e_i})\in\Delta'_{e_i}} [s(e_i,t_{e_i}) - s(e_i,t'_{e_i}) + \gamma]_+ \tag{9}$$

where $[x]_+ = max(0, x)$, and $\gamma > 0$ is the margin hyper-parameter. (e_i, t_{e_i}) and (e_i, t'_{e_i}) are the positive and negative training samples selected from the positive sample set Δ_{e_i} and the negative sample set Δ'_{e_i}, respectively. Here, the positive sample are entity type instances observed while the negative sample set is constructed by randomly selecting false types of the entity from the entire type set, i.e., $\Delta'_{e_i} = \{(e_i, t'_{e_i}) | t'_{e_i} \notin \mathcal{T}(e_i)\}$, where $\mathcal{T}(e_i)$ is the type set of the entity e_i.

4 EXPERIMENT

We study and evaluate our model on the entity type prediction task with two standard entity typing datasets. Further, we show the experimental results and some analysis of them.

Datasets		FB15k	YAGO43k
ET	#Entity	14,951	42,335
	#Type	3,851	45,182
	#Train	136,618	375,853
	#Valid	15,749	42,739
	#Test	15,780	42,750
KG	#Entity	14,951	42,335
	#Relation	1,345	37
	#Triplet	483,142	331,687

Table 1: Dataset statistics.

4.1 Experimental Setup

Dataset. We evaluate our model on two real-world KGs, FB15k [3] and YAGO43k [20], which are widely used in KG embedding literatures. FB15k and YAGO43k are the subsets of Freebase [2] and YAGO [29] respectively. Additionally, two supplementary entity type datasets FB15kET and YAGO43kET [20] are used to map entities of FB15k and YAGO43k to their entity types. To maintain fair comparison with baselines, we follow preprocessing steps of [20] and directly apply their split settings of train(80%), validation(10%), and test(10%) in our experiment. The basic statistics of datasets are shown in Table 1.

Baselines. In this paper, we select several state-of-the-art entity typing methods as our baselines.

- RESCAL [23], TransE [3], HOLE [22]. These are the typical embedding methods, which learn the embeddings of entities and their types from the triplet (entity, rdf:type, type).
- RESCAL-ET, TransE-ET, HOLE-ET [20]. These methods consider the contextual relations of a triplet to learn KG's embedding based on the models RESCAL, TransE, HOLE.
- HMGCN [13]. This model utilizes GCN [14] to capture semantic correlations between entities for typing prediction. For fair comparison, dependency among types is not used in HMGCN's training stage.
- ETE [20]. This model learns the embedding of each type to be closer to its entity embedding, and further makes type prediction.
- ConnectE [47]. This model makes the translating assumption on entity triplets and type triplets to learn their embeddings, and further encodes plausibility between the entity and its type.

Evaluation Protocol. In the testing phase, we first substitute the entity type t_e of each test sample (e, t_e) by every type $t \in \mathcal{T}$, and use the score function $s(e, t)$ to calculate the scores of these corrupted samples. We then rank these scores in ascending order and obtain the rank of original test sample (e, t_e). Inspired by TransE [3], the constructed negative sample maybe actually exist in KGs, also known as the false negative sample, which interferes with the final prediction performance. Hence we adopt the "filtered" setting [47] to filter the "false" negative samples before calculating score ranking. Following state-of-the-art baselines, we use the mean reciprocal rank (MRR) and HITS@{1,3,10}(%) as our metrics to evaluate our model.

DATASET	FB15kET				YAGO43kET			
METRICS	MRR	HITS@1	HITS@3	HITS@10	MRR	HITS@1	HITS@3	HITS@10
RESCAL[23]	0.19	9.71	19.58	37.58	0.08	4.24	8.31	15.31
RESCAL-ET[20]	0.24	12.17	27.92	50.72	0.09	4.32	9.62	19.40
HOLE[22]	0.22	13.29	23.35	38.16	0.16	9.02	17.28	29.25
HOLE-ET[20]	0.42	29.40	48.04	66.73	0.18	10.28	20.13	34.90
TransE[3]	0.45	31.51	51.45	73.93	0.21	12.63	23.24	38.93
TransE-ET[20]	0.46	33.56	52.96	71.16	0.18	9.19	19.41	35.58
ETE[20]	0.50	38.51	55.33	71.93	0.23	13.73	26.28	42.18
HMGCN[13]	0.51	39.02	54.75	72.36	0.25	14.21	27.34	43.69
ConnectE[47]	0.59	49.55	64.32	79.92	0.28	16.01	30.85	47.92
Ours	**0.62**	**51.66**	**67.68**	**82.13**	**0.35**	**24.43**	**41.31**	**56.48**

Table 2: Performance comparison on knowledge graph entity typing.

Model	FB15kET	YAGO43kET
AttEt-without-Neighbors	49.60%	17.13%
AttEt-without-Attention	49.87%	22.87%
AttEt	**51.66%**	**24.43%**

Table 3: Performance of different variants of AttEt on HITS@1.

Neighborhoods sampling. The entities in KGs usually have different numbers of neighborhoods. Also, not every neighborhood has a positive effect on type prediction. Thus, similar to the dropout technique, independent repeated sampling is executed for n times from the neighborhood set of the entity e_i to construct $\mathcal{N}(e_i)$.

Hyperparameters setting. In our model, the embeddings of the entities and relations are initialized by TransE [3], and the embeddings of entity types are initialized by uniform distribution $[-0.03, 0.03]$. In training stage, we select the learning rate $\mu \in \{0.0001, 0.0003, 0.001, 0.003, 0.01, 0.03, 0.1\}$, the smoothing factor $\lambda \in \{0, 0.2, 0.4, 0.6, 0.8, 1.0\}$, the margin $\gamma \in \{0, 1, 2\}$, the embedding dimension $d \in \{50, 100, 200\}$, the batch size $b \in \{16, 32, 64, 256\}$, and the size of neighborhood set $n \in$ {4, 8, 16, 32, 64, 128, 256, 512, 1024}. The optimal hyperparameter configuration is selected based on MRR by grid search in valid dataset as follows: the learning rate μ is 0.003, the margin γ is 1, the embedding dimension of entity, relation and entity type d are all set to 200, the batch size b is 64, the size of neighborhood set n is 128, the smoothing factor λ is 0.2 for FB15kET and 0.8 for YAGO43kET. We employ AdaGrad [7] to optimize the model's hyperparameters and run 800 epochs on FB15kET and YAGO43kET. Training our model on a machine with one Telsa-V100 graphics card takes about 19 hours on average.

4.2 Entity Type Prediction

The objective of this task is to complete a pair (entity, entity type) when its type is missing. Table 2 gives the convincing results of entity type prediction on FB15kET and YAGO43kET. From Table 2, we can observe that our model consistently outperforms all baselines on two datasets with all metrics. It confirms that our model successfully uses the neighborhood context of entities to improve the prediction performance on the entity typing task. That is, the type-specific attention mechanism can assign varying weight on neighborhoods of the entities, and further is indeed beneficial to reveal the fine-grained entity types. Specifically, for FB15kET, our model achieves 0.62 on MRR, 51.66% on HIST@1, 67.68% on HIST@3 and 82.13% on HIST@10, which is at least 0.03 on MRR, 2.11% on Hits@1, 3.36% on Hits@3 and 2.21% on Hits@10 higher performance than state-of-the-art methods. Similarly, for YAGO43kET, our model brings a 0.07 gain on MRR, 8.42% on Hits@1, 10.46% on Hits@3 and 8.56% on Hits@10 compared with baselines. Moreover, our model achieves higher performance improvements on YAGO43kET than FB15kET since YAGO43kET is sparser and contains a larger number of relation-sparsity entities than FB15kET. It suggests the smoothing factor λ can effectively strive for model's robustness when facing the relation-sparsity entities.

In order to qualitatively verify the effectiveness of our model, we construct the following variants of our model and report their results on HITS@1 metric:

- **AttEt without Neighbors.** This model infers entity types without considering the neighborhood of the entities. Similar to ConnectE [47], it directly uses the entity-level representations of the entities to calculate the matching score between the entity and its candidate type.
- **AttEt without Attention.** This model does not use the type-specific attention mechanism to learn the neighbor-level representation of the entities, which assigns the equal weight $\alpha'_{ij} = \frac{1}{n}$ on every neighborhood.

The results are reported in Table 3, which can be observed that AttEt achieves the best performance on entity type prediction among all variants. We attribute the superiority of AttEt to the following aspects: (1) AttEt improves by 2.06% and 7.3% on HITS@1 compared with AttEt-without-Neighbors. It suggests that AttEt successfully incorporates neighborhood context to enhance the prediction performance. This once again confirms that the neighborhood context of the entities indeed implies the fine-grained types. (2) AttEt has 1.79% and 1.56% higher performance on FB15kET and YAGO43kET than AttEt-without-Attention respectively, which indicates that the type-specific attention mechanism that assigns the type-specific weights on neighborhoods is beneficial to improve prediction performance.

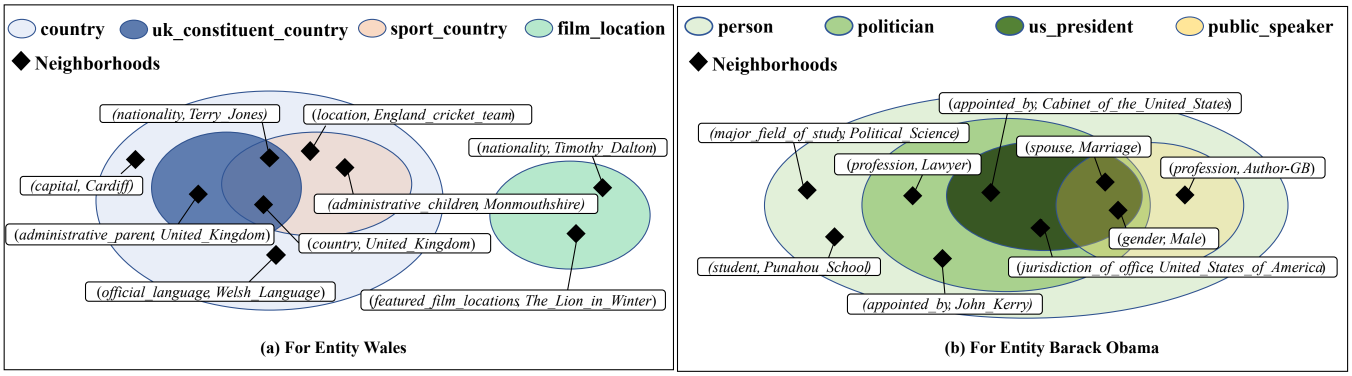

country
uk_constituent_country
sport_country
film_location
Neighborhoods
(nationality, Terry_Jones)
(location, England_cricket_team)
(nationality, Timothy_Dalton)
(capital, Cardiff)
(administrative_children, Monmouthshire)
(administrative_parent, United_Kingdom)
(country, United_Kingdom)
(official_language, Welsh_Language)
(featured_film_locations, The_Lion_in_Winter)
(a) For Entity Wales
person
politician
us_president
public_speaker
Neighborhoods
(appointed_by, Cabinet_of_the_United_States)
(major_field_of_study, Political_Science)
(spouse, Marriage)
(profession, Lawyer)
(profession, Author-GB)
(gender, Male)
(student, Punahou_School)
(jurisdiction_of_office, United_States_of_America)
(appointed_by, John_Kerry)
(b) For Entity Barack Obama

Figure 3: Fine-grained Entity Type in form of Venn Diagram

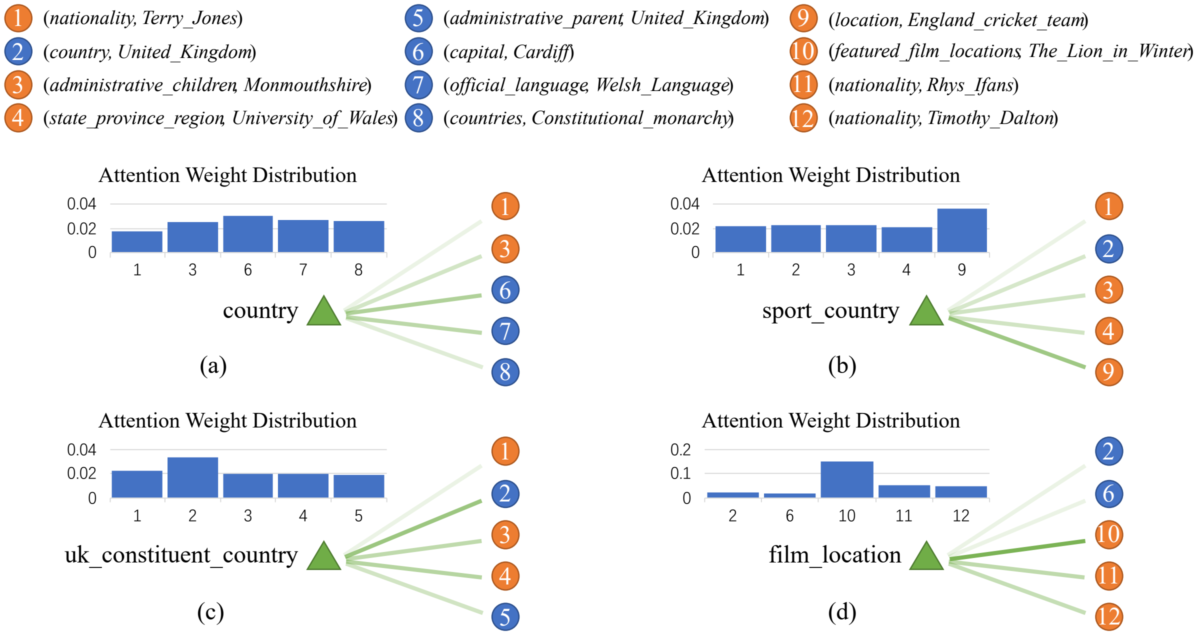


Figure 4: Attention weight visualization for entity *Wales*.

4.3 Effectiveness of Neighborhoods-Attention on Fine-grained Entity Type

In the real KGs, it usually exists correlation among the fine-grained types, such as subtype, intersection, and disjunction patterns. For example, the type */location/uk_statistical_location* is the subtype of the type *location/location*, the two types */location/uk_statistical_location* and */location/region* are intersected as such two entity types are overlapped or similar in some topic, and the type */tv/tv_writer* is disjoint with the type */location/location* as they are absolutely unrelated with each other in semantic.

In this paper, the type-specific attention mechanism can assign different weights on the neighborhood of the entity for the entity's different types, which can effectively confirm the hierarchical pattern among the fine-grained entity types. Intuitively, the neighborhoods with the larger weight can better reflect the semantic characteristics of the type. Thus, for each entity type of each entity, we rank the normalized attention score assigned on neighborhoods in decreased order, and select the top neighborhoods with higher scores where the sum of their score is just up to the threshold θ as Equation (10). $\theta = 0.5$ is empirically in our experiments.

$$\mathcal{N}'(t_{e_i}^k) = \{x_1, x_2, ..., x_m\} \tag{10}$$

where $t_{e_i}^k$ is the k-th type of the entity e_i, $x_j \in \mathcal{N}(e_i)$, $s_{x_j} \geq s_{x_{j+1}}$ and $\sum_{x \in \mathcal{N}'(t_{e_i}^k)} s_x \geq \theta$.

In this section, we define the selected top neighborhoods $\mathcal{N}'(t_{e_i}^k)$ as the support set of the type with a given entity. To obtain the correlation among types, we use the support set of the type as the bridge to verify the subtype, intersection, and disjunction relation patterns among types:

- **Subtype.** If the support set of type A is a subset of the support set of type B, then type A is a subtype of type B, i.e., type B is a supertype of type A. For example, the type */location/uk_civil_parish* is a subtype of the type */location/location*. This strict definition can not establish the subtype relation between type A and B even if just only one element of type A is not included in type B. In practice, the strict subset between two types rarely occurs and usually suffers from noise elements. To overcome this issue, we introduce a factor α and define a α-**Subtype** relation to make more flexible calculation as Equation (11). The α-Subtype can be established once α proportion of A is included by B instead of all elements.

$$A \subset_{\alpha} B \equiv \frac{|A \cap B|}{|A|} \geq \alpha \tag{11}$$

- **Intersection.** If the support set of type A and type B share at least one neighborhood, then the A is intersected with type B. For example, the two type */location/region* and */location/uk_statistical_location* are similar in some topics. The number of the shared neighborhoods (i.e., overlapping portion) between the support set of type A and type B can model the similarity between two types. The bigger the overlapping portion, the more similar they are. Practically, the duplicated or redundant types are usually introduced during constructing or populating the knowledge graphs, which makes it necessary to measure the similarity between the types. Here, we adopt α-Subtype to define the α-**Intersection** relation as Equation (12), where type A and type B are α-Intersection if they are α-Subtype of each other.

$$A \sim_{\alpha} B \equiv A \subset_{\alpha} B \text{ and } B \subset_{\alpha} A \tag{12}$$

- **Disjunction.** In contrast, entity types A and B are disjoint if their support set do not share any neighborhood element, which implies these two types focus on the different topics, such as */tv/tv_writer* and the */location/uk_statistical_location*.

Based on the definition of the above relation patterns among fine-grained entity types, we take typical type *location* and *person* as examples, and visualize the hierarchical graph in Figure 3 under the support set of types. In this experiment, we select the entity *Wales* and *Barack_Obama* as the target entities, and use them to confirm the hierarchical correlation among their fine-grained type labels. As shown in Figure 3. An oval that is completely contained in another oval is a subtype of that oval: any entity exhibiting the inner type will exhibit the outer one as well. Overlapping ovals like *country* and *sport_country* represent indicate that types are not mutually exclusive and intersected with each other. To some extent, from the examples in Figure 3, we can observe that: (1) The learned type-specific attention weights on neighborhoods can effectively reflect the subtype hierarchical structure among the fine-grained entity types. Specifically, the fine-grained types *uk_constituent_country* and *music/artist* are the subtype of the coarse type *country* and *person*, respectively. (2) The weight distributions on neighborhoods also imply the intersection correlation pattern between the entity types. For example, the two types *country* and *sport_country* are intersected with each other as they are similar in some neighborhoods. Similarly, the two types *us_president* and *public_speaker* are also intersected with each other. (3) The disjunction pattern is also revealed by the weight distributions on neighborhoods. In particular, the type *film_location* is disjoint with *country* as they have no overlapped neighborhood to support their type semantic.

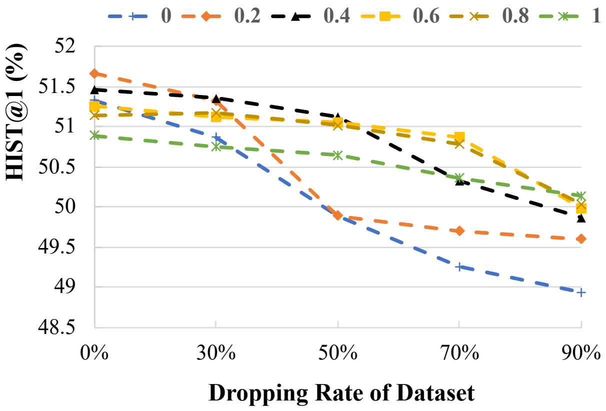


Figure 5: Performance with different smooth factor settings on FB15kET with different dropping rates.

To make the analysis more concrete and intuitive, we take the entity *Wales* and its several typical types from FB15KET as examples for the case study. For each candidate type of the entity *Wales*, its attention weight distribution over all neighborhoods is calculated as Equation (2), which implies the varying importance of different entity's neighborhoods to infer the entity type. For a more intuitive analysis, we select the top-5 neighborhood with larger attention weight of each type of the entity *Wales*, and visualize them in Figure 4. In this figure, the orange color indicates the incoming neighborhoods and blue is the outgoing neighborhoods. For each subfigure, the histogram displays the attention weight distribution over the top-5 neighborhoods, where y-axis is the attention weight score calculated from Equation (2) and x-axis are the indexes of the neighborhoods listed at the top. Besides, we also visualize these attention weights by the green color, where the thicker the green line, the larger the weight.

From Figure 4, we can see that: (1) For the same entity, the attention weight distribution over its neighborhood set is varying for its different types. Commonly, entity types always pay more attention to neighborhoods that reveal the semantic with them. For example, the type *uk_constituent_country* assigns higher weights on the neighborhood (*Wales, adjoins, England*) and (*Wales, administrative_parent, United_Kingdom*). Moreover, these neighborhoods with higher weights can slightly give clues about why does the entity *Wales* belongs to the type *uk_constituent_country*. (2) The attention weight distribution of subtype entity types is more concentrated while that of supertype entity types is more uniform. For example, the subtype type *uk_constituent_country* heavily centralizes on the neighborhoods (*Wales*, *country*, *United_Kingdom*) with larger attention weight. While the supertype *country* assigns the scattered weights with similar values on neighborhoods, like

(*Wales*, *capital*, *Cardiff*), (*Wales*, *official_language Welsh_language*) and (*Wales*, *countries*, *Constitutional_monarchy*). It again reveals that the type-specific weight distribution on neighborhoods can reveal the correlation between fine-grained entity types.

4.4 Smooth Factor over Attention

In practice, there are a large proportion of entities having sparse relations with other entities in KGs. To strive for our model's robustness when facing the relation-sparsity entity, we employ the smooth factor λ over type-specific attention mechanism as Equation (4). In this section, we discuss how different smooth factor setting affects the performance under different degree of sparsity of dataset.

To construct datasets with different degrees of sparsity, neighborhoods of entities in FB15KET are dropped randomly with different rates as Equation (13). In our experiment, the dropping rates $p \in \{$ 90%, 70%, 50%, 30%, 0% $\}$ are adopted, which means what percentage of neighborhoods to be dropped.

$$|\mathcal{N}'_p(e_i)| = (100\% - p) \times |\mathcal{N}(e_i)| \tag{13}$$

where $\mathcal{N}'_p(e_i) \subset \mathcal{N}(e_i)$ is the neighborhood set dropped from the original datasets $\mathcal{N}(e_i)$ with dropping rate p. Particularly, the dropping rate 0% is the baseline where the original dataset is used directly without dropping in practice.

Based on optimal settings described in Section 4.1, we report the results with different smooth factors $\lambda \in \{$ 0.0, 0.1, 0.3, 0.5, 0.7, 0.9, 1.0 $\}$ on all datasets with different degrees of sparsity, i.e., under different dropping rate p on the original dataset. Particularly, when λ is 1, Equation (4) disables the attention mechanism and $\alpha'_{i,j}$ equals $\frac{1}{n}$, which is a linear combination over neighborhoods to obtain the neighbor-level features. While λ is 0, it is fully weighted by the attention score. The λ between 0.0 and 1.0 is a trade-off between the attention mechanism and the linear combination. Analyzing such performance comparison shown in Figure 5, we have the following observations: (1) The optimal smooth factor depends on the degree of sparsity of the dataset. For the sparser data sets, the performance of the model is better by choosing larger λ. It means putting more weight on the average of the neighborhoods. Particularly, the dataset with a 90% dropping rate makes a higher performance with $\lambda = 1.0$ than that with $\lambda = 0.0$ about 1.2%. (2) The degree of sparsity of dataset indeed affect performance directly. Compared with the baseline (p = 0%), the best hit@1 measure under $p = \{30\%, 50\%, 70\%, 90\%\}$ drops to 0.31%, 0.54%, 0.79%, 1.53% respectively, which suggests that the sparser dataset is harder to achieve higher performance than the dense one.

5 RELATED WORK

With the widespread application of KGs, knowledge graph completion (KGC) becomes increasingly important in both academia and industry. Most existing approaches like translation-based methods [3, 21, 45] mainly focus on performing link prediction task and triplet classification task, which predicts the missing h or t for a golden triplet (h, r, t) and judges whether a given triplet (h, r, t) is correct or not, respectively. For example, the typical method TransE [3] treats the relation as a translation vector to link the head and tail entities, and makes the triplet satisfy $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$. Later translation-based works such as [11, 17, 19, 35, 36] are presented to learn the KG's embeddings. Also, there exists many semantic-matching methods [24, 30–32, 42] that employ the matching mechanism to explore the plausibility of the triplet, such as Dismult [42] defines the matching operation as $\mathbf{h}\mathbf{M}_r\mathbf{t} \approx 1$ to learn KG's embeddings.

Fousing on the entity typing task, the previous studies [4, 5, 25, 38, 40, 41, 44] usually mine fine-grained entity type with external auxiliary information, such as web search query logs [26], the textual surface patterns [43], and Wikipedia [48]. These methods have low generality and flexibility as they are limited by the quality of auxiliary information. Recently, the structured facts in KGs can provide more precise and solid information in inferring entity types. A number of methods use KGs to complete the missing entity types have been proposed. One method [20, 47] attempts to learn the entity embeddings from KGs by representation learning methods [3], and further build the transmission between entities and their types. For example, ConnectE [47] employs the entity triplet and type triplet together to learn the embeddings of entities and types, and project the entity embeddings into type space to make type inference. The second method [12, 27, 39, 41] utilizes entity's feature (textual description, property and category) or mention in text to do inference. For instance, HMGCN [13] uses GCN to embed multiple kinds of semantic correlations between entities to make typing prediction. However, these work ignore the different importance of neighborhood on inferring different entity types. To address the issues, we propose a neighborhood-attention neural network, which enables the fine-grained entity types to have their own weight distribution on neighborhoods for capturing their characteristics.

6 CONCLUSION

In this paper, we propose a neighborhood-attention neural fine-grained entity typing model AttEt for KG completion, which incorporates the neighborhoods of the entity to bridge the entities and their types together. Specifically, AttEt devises a type-specific attention mechanism on the neighborhoods for learning the neighbor-level representation of the entities, and adaptively incorporates the embeddings of entities with their neighbor-level representation to make entity type inference. The varying attentive scores on neighborhoods of the entity can imply and reveal the complex correlation between the fine-grained entity types. Besides, AttEt develops the smooth factor to strive for its robustness when facing the relation-sparsity entities. The experimental results on two datasets show that AttEt significantly outperforms state-of-the-art baselines in the HITS@1 by 2.11% on Freebase and by 8.42% on YAGO.

ACKNOWLEDGMENTS

This work was supported by the project BMKY2020B10, the National Natural Science Foundation of China (Grant No. 62102421), and Intelligent Social Governance Platform, Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative, Renmin University of China. I also wish to acknowledge the support provided and contribution made by Public Policy and Decision-making Research Lab of RUC.

REFERENCES

[1] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase:a collaboratively created graph database for structuring human knowledge, In In Proceedings of KDD. *In processings of the 2008 ACM SIGMOD international Conference on Management of Data*, 1247–1250.

[2] Kurt D. Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Sigmod Conference*.

[3] A. Bordes, N. Usunier, and A. Garcia-Dur!'äan. 2013. Translating Embeddings for Modeling Multi-relational Data. In *In Proceedings of NIPS*. 2787–2795.

[4] Tongfei Chen, Yunmo Chen, and Benjamin Van Durme. 2020. Hierarchical Entity Typing via Multi-level Learning to Rank. In *Proceedings of ACL*. Association for Computational Linguistics, Online, 8465–8475.

[5] Eunsol Choi, Omer Levy, Yejin Choi, and Luke Zettlemoyer. 2018. Ultra-Fine Entity Typing. In *Proceedings of ACL*. Association for Computational Linguistics, Melbourne, Australia, 87–96.

[6] Wanyun Cui, Yanghua Xiao, Haixun Wang, Yangqiu Song, Seung-won Hwang, and Wei Wang. 2017. KBQA: learning question answering over QA corpora and knowledge bases. *Proceedings of the VLDB Endowment* 10, 5 (Jan. 2017), 565–576. https://doi.org/10.14778/3055540.3055549

[7] John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research* 12, 7 (2011).

[8] Hannaneh Hajishirzi, Leila Zilles, Daniel S. Weld, and Luke Zettlemoyer. 2013. Joint Coreference Resolution and Named-Entity Linking with Multi-Pass Sieves. In *Proceedings of EMNLP*. Association for Computational Linguistics, Seattle, Washington, USA, 289–299.

[9] Xiao Huang, Jingyuan Zhang, Dingcheng Li, and Ping Li. 2019. Knowledge Graph Embedding Based Question Answering. In *Proceedings of WSDM (WSDM '19)*. Association for Computing Machinery, New York, NY, USA, 105–113.

[10] Prachi Jain, Pankaj Kumar, Soumen Chakrabarti, et al. 2018. Type-sensitive knowledge base inference without explicit type supervision. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 75–80.

[11] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao. 2015. Knowledge graph embedding via dynamic mapping matrix. In *In Proceedings of ACL*. 687–696.

[12] Hailong Jin, Lei Hou, Juanzi Li, and Tiansi Dong. 2018. Attributed and Predictive Entity Embedding for Fine-Grained Entity Typing in Knowledge Bases. In *Proceedings of ACL*, Emily M. Bender, Leon Derczynski, and Pierre Isabelle (Eds.). Association for Computational Linguistics, 282–292.

[13] Hailong Jin, Lei Hou, Juanzi Li, and Tiansi Dong. 2019. Fine-Grained Entity Typing via Hierarchical Multi Graph Convolutional Networks. In *Proceedings of EMNLP*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, 4968–4977.

[14] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of ICLR*. OpenReview.net.

[15] J. Lehmann, R. Isele, and M. Jakob. 2015. DBpedia: A large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web* 6, 2 (2015), 167–195.

[16] Jian Li, Yong Liu, Rong Yin, Hua Zhang, Lizhong Ding, and Weiping Wang. 2018. Multi-Class Learning: From Theory to Algorithm. *NeurIPS* 31 (2018), 1593–1602.

[17] Yankai Lin, Zhiyuan Liu, Xuan Zhu, Xuan Zhu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of AAAI*. 2181–2187.

[18] Shuman Liu, Hongshen Chen, Zhaochun Ren, Yang Feng, Qun Liu, and Dawei Yin. 2018. Knowledge Diffusion for Neural Dialogue Generation. In *Proceedings of ACL*. Association for Computational Linguistics, Melbourne, Australia, 1489–1498.

[19] Yong Liu, Shizhong Liao, Shali Jiang, Lizhong Ding, Hailun Lin, and Weiping Wang. 2020. Fast Cross-Validation for Kernel-Based Algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42, 5 (2020), 1083–1096. https://doi.org/10.1109/TPAMI.2019.2892371

[20] Changsung Moon, Paul Jones, and Nagiza F Samatova. 2017. Learning entity type embeddings for knowledge graph completion. In *Proceedings of CIKM*. 2215–2218.

[21] Dai Quoc Nguyen, Thanh Vu, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Phung. 2019. A Capsule Network-based Embedding Model for Knowledge Graph Completion and Search Personalization. In *Proceedings of NAACL*. Association for Computational Linguistics, Minneapolis, Minnesota, 2180–2189.

[22] Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. 2016. Holographic Embeddings of Knowledge Graphs. In *Proceedings of AAAI (AAAI'16)*. AAAI Press, 1955–1961.

[23] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data.. In *Proceedings of ICML*, Vol. 11. 809–816.

[24] Maximilian Nickel, Volker Tresp, and Hans Peter Kriegel. 2011. A Three-Way Model for Collective Learning on Multi-Relational Data. In *International Conference on Machine Learning, ICML 2011*. 809–816.

[25] Yasumasa Onoe, Michael Boratko, and Greg Durrett. 2021. Modeling Fine-Grained Entity Types with Box Embeddings. *CoRR* abs/2101.00345 (2021).

[26] Patrick Pantel, Thomas Lin, and Michael Gamon. 2012. Mining Entity Types from Query Logs via User Intent Modeling. In *Proceedings of ACL*. Association for Computational Linguistics, Jeju Island, Korea, 563–571.

[27] Heiko Paulheim and Christian Bizer. 2013. Type Inference on Noisy RDF Data. In *Proceedings of the 12th International Semantic Web Conference - Part I (ISWC '13)*. Springer-Verlag, 510–525. https://doi.org/10.1007/978-3-642-41335-3_32

[28] Qingyu Guo, Fuzhen Zhuang, Chuan Qin, Hengshu Zhu, Xing Xie, Hui Xiong, and Qing He. 2020. A Survey on Knowledge Graph-Based Recommender Systems. *arXiv:2003.00911 [cs, stat]* (Feb. 2020). http://arxiv.org/abs/2003.00911 arXiv: 2003.00911.

[29] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of WWW*. 697–706.

[30] Théo Trouillon and Maximilian Nickel. 2017. Complex and Holographic Embeddings of Knowledge Graphs: A Comparison. *CoRR* abs/1707.01475 (2017).

[31] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex Embeddings for Simple Link Prediction. In *Proceedings of ICML*. 2071–2080.

[32] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2017. Graph Attention Networks. *CoRR* abs/1710.10903 (2017).

[33] Hongwei Wang, Hongyu Ren, and Jure Leskovec. 2020. Entity Context and Relational Paths for Knowledge Graph Completion. *CoRR* abs/2002.06757 (2020). arXiv:2002.06757

[34] Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. 2018. Explainable Reasoning over Knowledge Graphs for Recommendation. *arXiv:1811.04540 [cs]* (Nov. 2018). http://arxiv.org/abs/1811.04540 arXiv: 1811.04540.

[35] Z. Wang, J. Zhang, J. Feng, and Z. Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *In Proceedings of AAAI*. 1112–1119.

[36] H. Xiao, M. Huang, and X. Zhu. 2016. TransG: A Generative Model for Knowledge Graph Embedding. In *Proceedings of ACL*. 2316–2325.

[37] Ruobing Xie, Zhiyuan Liu, Jia Jia, Huanbo Luan, and Maosong Sun. 2016. Representation Learning of Knowledge Graphs with Entity Descriptions. In *Proceedings of AAAI*. AAAI Press, 2659–2665.

[38] Ji Xin, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2018. Improving Neural Fine-Grained Entity Typing With Knowledge Attention. In *Proceedings of AAAI*, Sheila A. McIlraith and Kilian Q. Weinberger (Eds.). AAAI Press, 5997–6004.

[39] Bo Xu, Yi Zhang, Jiaqing Liang, Yanghua Xiao, Seung-won Hwang, and Wei Wang. 2016. Cross-Lingual Type Inference. In *Proceedings of DASFAA (Lecture Notes in Computer Science)*, Vol. 9642. Springer, 447–462.

[40] Peng Xu and Denilson Barbosa. 2018. Neural Fine-Grained Entity Type Classification with Hierarchy-Aware Loss. In *Proceedings of ACL*. Association for Computational Linguistics, New Orleans, Louisiana, 16–25.

[41] Yadollah Yaghoobzadeh, Heike Adel, and Hinrich Schütze. 2018. Corpus-Level Fine-Grained Entity Typing. *J. Artif. Intell. Res.* 61 (2018), 835–862.

[42] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. *CoRR* abs/1412.6575 (2014).

[43] Limin Yao, Sebastian Riedel, and Andrew McCallum. 2013. Universal schema for entity type prediction. In *Proceedings of workshop on Automated knowledge base construction (AKBC '13)*. Association for Computing Machinery, New York, NY, USA, 79–84.

[44] Dani Yogatama, Daniel Gillick, and Nevena Lazic. 2015. Embedding Methods for Fine Grained Entity Type Classification. In *Proceedings of ACL*. Association for Computational Linguistics, Beijing, China, 291–296.

[45] Chuxu Zhang, Huaxiu Yao, Chao Huang, Meng Jiang, Zhenhui Li, and Nitesh V. Chawla. 2020. Few-Shot Knowledge Graph Completion. *Proceedings of AAAI Conference on Artificial Intelligence* 34, 03 (April 2020), 3041–3048. Number: 03.

[46] Richong Zhang, Fanshuang Kong, Chenyue Wang, and Yongyi Mao. 2018. Embedding of Hierarchically Typed Knowledge Bases. In *Proceedings of AAAI*, Sheila A. McIlraith and Kilian Q. Weinberger (Eds.). AAAI Press, 2046–2053.

[47] Yu Zhao, Anxiang Zhang, Ruobing Xie, Kang Liu, and Xiaojie Wang. 2020. Connecting Embeddings for Knowledge Graph Entity Typing. In *Proceedings of ACL*. Association for Computational Linguistics, Online.

[48] Ben Zhou, Daniel Khashabi, Chen-Tse Tsai, and Dan Roth. 2018. Zero-Shot Open Entity Typing as Type-Compatible Grounding. In *Proceedings of ACL*. Association for Computational Linguistics, Brussels, Belgium, 2065–2076.