This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2021.3070317, IEEE Transactions on Knowledge and Data Engineering

Active Learning for Knowledge Graph Schema Expansion

Seungmin Seo, Byungkook Oh, Eunju Jo, Sanghak Lee, Dongho Lee, Kyong-Ho Lee, Donghoon Shin, and Yeonsoo Lee

Abstract—Both entity typing and relation extraction from text corpora are widely used to identify the semantic types of an entity and a relation in a knowledge graph (KG). Most existing approaches rely on a pre-defined set of entity types and relation types in a KG. They thus cannot map entity mentions (relation mentions) to unseen entity types (relation types). To fundamentally overcome the limitations, we should add new semantic types of entities and relations to a KG schema. However, schema expansion traditionally requires manual conceptualization through a user's observation on the text corpus while assuming the existence of suitable target KG schemas. In this work, we propose an **A**ctive learning framework for **K**nowledge graph **S**chema **E**xpansion (**AKSE**), which can generate a new semantic type for KG schemas, without depending on a set of target schemas and human users' observation. Specifically, a granularity based active learning algorithm determines whether a KG schema requires new semantic types or not. We also introduce a KG schema attention-based neural method which assigns semantic types to the entities and relationships extracted. To the best of our knowledge, our work is the first study to expand a KG schema with active learning.

Index Terms—Knowledge Graph, Active Learning, Knowledge Graph Schema Expansion, Relation Extraction.

1 Introduction

KNOWLEDGE graphs (KGs) are useful resources for many natural language processing tasks, including entity recognition, relation extraction, and question answering [1]. KGs are organized with a set of facts (i.e., triples), which consist of the entities connected by a relation, and a set of semantic types in a hierarchical schema defining the meaning of instances [2]. For example, in the ⟨*Donald Trump, is_ president_ of, USA*⟩ triple, *Donald Trump* is an entity of the *(Object/Agent/Person/Politician)* type, and *is_ president_ of* is a relation type between *Donald Trump* and *USA*.

Extracting entities and relations from text is the task of assigning types to entity mentions and finding a relation type between two entity mentions, which can be considered as the label classification problem of KGs. Typing entities and relations usually requires a large amount of human-annotated corpora with the help of human expertise. Thus, existing methods mostly depend on distant supervision to obtain annotated sentences, which can automatically generate training data by aligning facts in KG with sentences in texts [3]. However, a distantly supervised corpus contains a tremendous number of errors since only the database supervises the types of entities and relations. Moreover, since existing approaches rely on a KG schema with a fixed set of types, they fail to assign the unseen types detected in corpora to entities and relations (see Figure 1).

To fundamentally address the above issue, unseen semantic types need to be conceptualized and added to a KG schema. However, the prior approaches for schema expansion have depended on manual conceptualization through a user's precise observation of the text corpus and the existence of target schemas [4]. In this paper, we propose an **A**ctive learning framework for **K**nowledge graph **S**chema **E**xpansion (**AKSE**). As shown in Figure 2, given a text corpus and an initial KG schema, our framework first predicts the semantic types of detected entities and relationships via the proposed convolutional neural model employing KG schema hierarchy attention. In order to identify the semantic types needing expansion, we propose a granularity score based active learning method; a semi-supervised machine learning algorithm, which interactively queries human users to obtain ground-truth training data (i.e., gold triples) incrementally. The proposed framework can evolve an initial KG schema without domain expertise and the massive amount of annotated text corpora. Our method is general and applicable to a wide range of text in any domain.

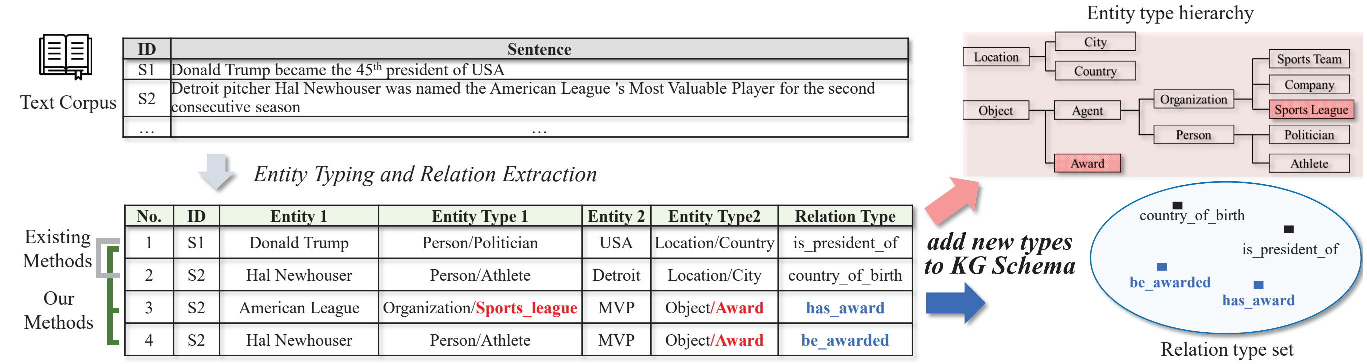
We study the validity of the proposed approach with three sets of experiments. 1) To estimate the performance of schema expansion, we construct a seed KG schema and a target KG schema derived from the Freebase[1] type system. Then we compare our framework with the state-of-the-art method through simulation and human annotation experiments. 2) To verify the efficacy of the annotation process in the proposed framework, we compare the proposed neural model with state-of-the-art entity typing and relation extraction methods. 3) To confirm that computing the granularity score is critical for selecting the types that need to be expanded, we compare the sampling strategies with three active learning baselines. The experimental results show

• S. Seo, B. Oh, E. Jo, S. Lee, D. Lee, and K.-H. Lee are with the Department of Computer Science, Yonsei University, Seoul, Republic of Korea, 03722. E-mail: smseo91@yonsei.ac.kr, bkoh509@gmail.com, {ejjo, shlee, dhlee}@icl.yonsei.ac.kr, khlee89@yonsei.ac.kr
D. Shin and Y. Lee are with the NCSOFT NLP Center, Republic of Korea, 13494
E-mail: dhshin@ncsoft.com, yslee@ncsoft.com

Manuscript received ...
(Corresponding author: Kyong-Ho Lee)

1. https://developers.google.com/freebase/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2021.3070317, IEEE Transactions on Knowledge and Data Engineering



ID	Sentence
S1	Donald Trump became the 45th president of USA
S2	Detroit pitcher Hal Newhouser was named the American League 's Most Valuable Player for the second consecutive season
…	…

No.	ID	Entity 1	Entity Type 1	Entity 2	Entity Type2	Relation Type
1	S1	Donald Trump	Person/Politician	USA	Location/Country	is_president_of
2	S2	Hal Newhouser	Person/Athlete	Detroit	Location/City	country_of_birth
3	S2	American League	Organization/**Sports_league**	MVP	Object/**Award**	**has_award**
4	S2	Hal Newhouser	Person/Athlete	MVP	Object/**Award**	**be_awarded**

Fig. 1. Existing methods cannot assign unseen types (*Sports_league*, *Award*) to entities (*AmericanLeague*, *MVP*), nor extract relation types (*has_ award*, *be_ awarded*), if there are no such types in the KG schema. Our proposed method overcomes these limitations via an active learning.

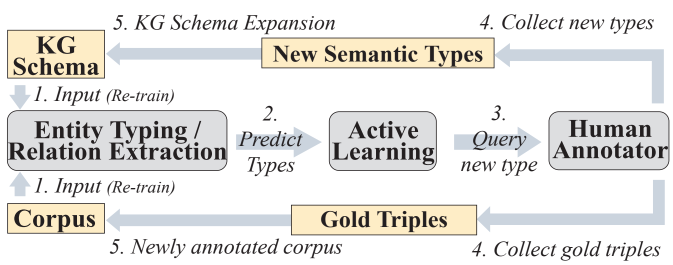


Fig. 2. Active learning framework for knowledge graph schema expansion.

that the proposed approach expands a KG schema more accurately than the state-of-the-art methods.

The major contributions of our work are stated as follows.

- We propose a novel framework, which expands a KG schema from a seed schema with a given text corpus, not relying on domain expertise.
- We introduce a neural model that can carry out both entity typing and relation extraction with the help of the KG schema attention mechanism.
- We present an active learning algorithm using a granularity score as an uncertainty measure to select the semantic type to be expanded.
- We conduct experiments on various tasks to estimate the validity of our approach.

The rest of the paper is organized as follows. Section 2 describes related works and Section 3 defines the problem. Neural models for entity typing and relation extraction are introduced in Section 4. A granularity score based active learning algorithm is given in Section 5. Section 6 describes the experimental results and Section 7 concludes the paper.

2 RELATED WORK

The proposed method is primarily related to three strains of research: entity typing, relation extraction, and active learning. This section summarizes these research directions.

2.1 Entity Typing

Entity typing is the task of classifying a recognized entity in a text sentence into a semantic type in the entity type hierarchy. Entity typing traditionally handles coarse-grained entity types such as a *person*, *location*, and *organization*. Recently, numerous researchers have aimed to label an entity with more fine-grained entity type sets organized in a hierarchy.

Yogatama et al. [5] use word embeddings to translate user-defined features into a low-dimensional vector space. Dong et al. [6] introduce a hybrid neural model, which only uses word features. Shimaoka et al. [7] attempt to incorporate the context of entity mentions with an attention-based neural network model. Further, Shimaoka et al. [8] consider hand-crafted features and combine them with learned features. Ren et al. [9] propose an embedding method AFET, which separately models clean and noisy mentions in a text and incorporates type hierarchy to optimize the model. Following the idea from AFET, Ren et al. [10] jointly embed entity mentions, text features, and entity types into a low-dimensional space. Abhishek et al. [11] use label noise information in a variant of non-parametric hinge loss function for entity typing. More recently, Xu et al. [12] and Murty et al. [13] introduce methods for integrating hierarchical information, which optimize loss functions in order to improve entity typing. Interestingly, Zhong et al. [14] indicate the problem of position-based tagging scheme suffering from inconsistent tag assignment. They propose a constituent-based tagging scheme to recognize named entity types with time expressions.

However, none of the methods above assume the incompleteness of KG schemas where unseen entity types should be recognized in a text corpus. Thus they may not classify detected mentions into exact types. Although Ma et al. [15] proposed a prototype-driven label embedding method working for both seen and unseen types, and Huang et al. [16] presented an unsupervised entity typing method that requires no pre-defined entity types, those methods assume that target type sets already exist in external KG schemas and use that information while learning.

2.2 Relation Extraction

Relation extraction is the task of extracting relationships between entity mentions. Most prior approaches rely on existing entity recognizers or entity typing systems. In this work, we focus on the models that integrate entity typing and relation extraction.

Augenstein et al. [17] introduce a learning method of jointly training the named entity classifiers and the relation extractor. However, it is dependent on existing NLP tools like entity recognizer. More recently, Yaghoobzadeh et al. [18] introduce multi-instance multi-label learning algorithms to perform entity typing and relation extraction without existing entity typing tools. CoType [19] runs a text segmentation algorithm to extract entities and jointly embeds entity mentions, relation mentions, and type hierarchy. Di et al. [20] introduce a method to extract a relation via domain-aware transfer learning (ReTrans). Takanobu et al. [21] propose a hierarchical reinforcement learning framework (HRL) for relation extraction. To extract overlapped and discrete relations, Zhang et al. [22] propose a multi-labeled relation extraction method that employs an attentive capsule network. The method finds low-level capsules that contain related relation features, and then clusters them to represent high-level features. Singh and Bhatia [23] define the relationships derived directly from a sentence as the first-order relation. The target relation is calculated by combining the first-order relation scores by referring to the indirect relationship connected by the context token as the second-order relation.

However, like the prior works on entity typing, existing relation extraction methods assume that a target KG schema is ready and complete. This assumption would lead to mapping detected entities and relations into inaccurate semantic types.

2.3 Active Learning

Active learning enables a learning model to query human users to get the desired outputs interactively. Nowadays, active learning is widely used in various NLP tasks, such as entity extraction [24], [25], semantic role labeling [26], and named entity recognition [27].

Kolghi et al. [24] adopt active learning to annotate clinical concepts and reduce the time required to annotate domain concepts manually. Al et al. [25] propose an entity set expansion framework to annotate sparse entities rapidly in unstructured corpora. Wang et al. [26] present an active learning algorithm for black-box semantic role labeling. Zhang et al. [28] explore deep active learning for sentence classification with the convolutional neural network. Recently, Shen et al. [27] propose a deep active learning framework for named entity recognition with a CNN-CNN-LSTM architecture where the amount of training data is drastically reduced when deep learning is combined with active learning. Peshterliev et al. [29] propose a majority-CRF algorithm to select informative samples for improving the accuracy of new domains in a natural language understanding system. Hu et al. [30] introduce active learning with partial feedback, where the learner actively chooses both the example to label and the binary question to ask. Chaudhary et al. [31] use cross-lingual transfer learning to improve low-resource NER and employed active learning to select informative data for human annotators. Kasai et al. [32] introduce a deep learning-based method that targets low-resource settings for entity resolution using a novel combination of transfer learning and active learning. Li et al. [33] propose a discrete annotation method and asked annotators to identify mention antecedents for coreference resolution.

We believe that our work is the first study to expand a KG schema with active learning.

3 Background and Problem

3.1 Entity Mention, Relation, and Triple

Entity mention m is a string in a text, which is represented with entity e in knowledge graph Ψ. We assume that exactly one entity type $t \in \mathcal{T}$ is assigned to each entity. Thus, in this work, the probability that entity e has type t is equal to the probability that mention m has type t. A relation describes a type of relationship $r \in \mathcal{R}$ between a pair of entities $z = (e_1, e_2)$. We assume that each entity pair z has exactly one relation type $r(z)$. In a KG, a fact is represented with a triple $\langle e_1, r, e_2 \rangle$.

3.2 Knowledge Graph with Schema

A KG is a graph-based structured knowledge representation based on the RDF standard[2] [34]. In a KG, its schema is expressed with the same syntax and data model as the data itself. The sets of possible entity types (i.e., *classes*) and relation types (i.e., *properties*) are organized in a type hierarchy defining the interrelations and restrictions of their usage. We define a KG schema $\mathcal{S}$ as a set of an entity type hierarchy $\mathcal{T}$ and a relation type set $\mathcal{R}$. Specifically, an entity type hierarchy $\mathcal{T}$ is defined as a tree, which provides a way to classify and organize entities in a KG.

3.3 Problem Definition

Given a corpus $\mathcal{D}$ within a particular domain and a KG Ψ with an initial schema $\mathcal{S}^{\Psi}_{seed}$, which consists of a target entity type hierarchy $\mathcal{T}$, and a target relation type set $\mathcal{R}$, we aim to (1) extract entity types and relation types from a text corpus $\mathcal{D}$, (2) predict the types that need to be expanded, (3) query an annotator to generate a new semantic type, and (4) add unseen types to a KG schema $\mathcal{S}^{\Psi}_{seed}$. The output is an expanded KG schema $\mathcal{S}^{\Psi}_{expanded}$, where new semantic types are added.

4 Neural Model for Entity Typing and Relation Extraction

In this section, we introduce the neural model for predicting semantic types of entities and a relation type between entities. As shown in Figure 3, the proposed neural model embeds both sentences and a KG schema into a low-dimensional vector space. We use those representations to determine the most probable types of entities and relationships.

2. http://www.w3.org/RDF/

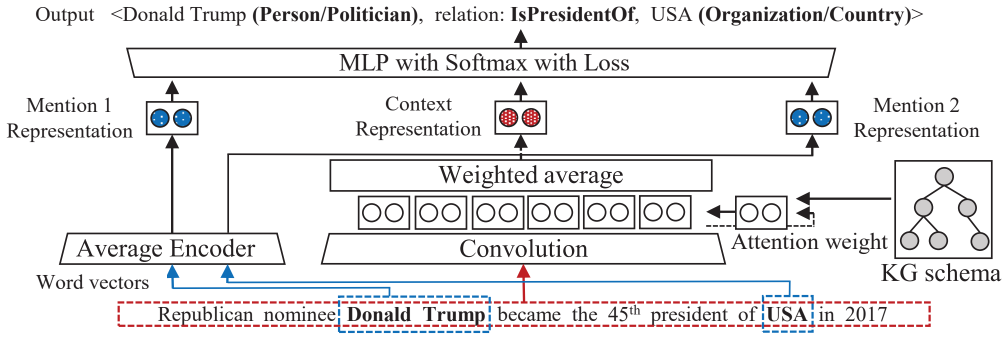


Fig. 3. An illustration of the proposed neural model predicting semantic types for entities and relations.

4.1 Sentence Encoder

Given a sentence s, our model firstly transforms each word into a vector representation via a word embedding matrix. These vectors are used to predict the entity types of the entity mentions detected and the relation type between entities. Specifically, we also incorporate word position embeddings which have been used to effectively reflect relative distances between the i^{th} words and their target entities [13], [12]. We concatenate the word embeddings and position embeddings, which are denoted as $\mathbf{w} = \{\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_n\}$, where $\mathbf{w}_i \in \mathbb{R}^d$ and n is the number of words in a sentence.

4.1.1 Mention Representation

Following the previous study [7], since a complex neural model like CNN or RNN tends to overfit, we adopt a simple averaging encoder for mention representation. A mention representation $\mathbf{m} \in \mathbb{R}^d$ is computed by averaging the word vectors of entity mention words $m_1, m_2, ...m_{n_m}$:

$$\mathbf{m} = \frac{1}{n_m}\sum_{i=1}^{n_m} \mathbf{m}_i, \tag{1}$$

where n_m is the length of an entity mention.

4.1.2 Context Representation

We adopt a single layer Attention-based Convolutional Neural Network (ABCNN) with a tanh non-linearity to represent contexts of a sentence [35]. We compute the i^{th} element of context vector as:

$$\mathbf{c}_i = \sum_{j=0}^{w}(\mathbf{F}[j] * \mathbf{q}[i - [\frac{w}{2}] + j]) + \mathbf{b}, \tag{2}$$

where $\mathbf{F} \in \mathbb{R}^{d\times d}$ is the CNN filter, $\mathbf{q} \in \mathbb{R}^d$ is the word vector of each context word, i indicates $0 \leq i \leq n - w + 1$, $\mathbf{b} \in \mathbb{R}^d$ is the bias, and the context window size w is set to 5.

In this work, we compute attention weights on the output of convolution rather than on the input representation. The main reason is that we can reduce the number of parameters and that the model is less vulnerable in overfitting. $\mathbf{A} \in \mathbb{R}^{d\times d}$ denotes the attention matrix, and $\mathbf{a}_i \in \mathbb{R}^d$ is the attention weight of $\mathbf{c}_i$, which will be introduced in the next section. The final context representation $\mathbf{c}$ is formed by a weighted sum of the pooling outputs:

$$\mathbf{c} = \sum \mathbf{a}_i * \mathbf{c}_i. \tag{3}$$

$$\mathbf{a}_i = f(\mathbf{t} * \mathbf{W}_a * \mathbf{c}_i), \tag{4}$$

where $f()$ denotes a non-linear function, $\mathbf{t}$ is the vector representation of an entity type generated by the knowledge graph schema encoder, and $\mathbf{W}_a$ corresponds to a bilinear parameter matrix.

Equation 4 shows that we directly utilize the correlation between the schema information of an entity mention in the sentence and the context of the sentence. The calculated attention weight $\mathbf{a}_i$ is used to generate final context representation $\mathbf{c}$ as shown in Equation 3.

4.2 Knowledge Graph Schema Encoder

The entity types in a KG schema have a hierarchical structure, where a lower-level entity type and an upper-level entity type form an antisymmetric relationship. It means that a lower-level entity type can be connected to an upper-level entity type by is_A relation, but the opposite relationship is not allowed. For example, while ⟨*Politician, is_A, Person*⟩ exists in the schema, ⟨*Person, is_A, Politician*⟩ does not.

To reflect the structural information of a KG schema when we predict the types of entities and relationships, we encode a KG schema via the **ComplEx** Bilinear model [36]. Through using Hermitian inner products, we map an entity type $\mathbf{t}$ to a complex-valued vector and a relation type $\mathbf{r}$ to a complex matrix. In particular, ComplEx is arguably simpler than other KG embedding methods. When the embedding dimension is K, the space and time complexity of ComplEx are $O(K)$. We compute the score used to predict the hierarchical structure between entity types as follows:

$$\begin{aligned} s(t_1, t_2) &= Re(\langle \mathbf{t}_1, \mathbf{r}, \mathbf{t}_2 \rangle) \\ &= Re(\sum_k \mathbf{t}_1 * \mathbf{r}_k * \mathbf{t}_2) \\ &= \langle Re(\mathbf{t}_1), Re(\mathbf{r}), Re(\mathbf{t}_2) \rangle \\ &+ \langle Re(\mathbf{t}_1), Re(\mathbf{r}), Im(\mathbf{t}_2) \rangle \\ &+ \langle Im(\mathbf{t}_1), Re(\mathbf{r}), Im(\mathbf{t}_2) \rangle \\ &- \langle Im(\mathbf{t}_1), Im(\mathbf{r}), Re(\mathbf{t}_2) \rangle, \end{aligned} \tag{5}$$

where $s()$ is the scoring function, Re is the real part of the decomposition generated by an affine transformation, Im is the imaginary part, r_k is one of the relation types between entity types t_1 and t_2, and k is the number of relation types that can exist between t_1 and t_2.

A hierarchical entity type can also be embedded by the *is_A* relation type $\mathbf{r}_{IsA}$, since the scoring function is antisymmetric as noted in [13]. In particular, we only encode the knowledge graph schema (i.e. T-box triples), not the complete knowledge graph (i.e. T-box and A-box triples). For example, we only encode the triple ⟨*Politician, isPresidentOf, Country*⟩, not the triple ⟨*Donald Trump, IsPresidentOf, USA*⟩.

4.3 Type Predictor

We use a two-layer Multi-Layer Perceptron (MLP) to learn predict functions $P(t|e)$ and $P(r|z)$, which compute the probability that an entity e has an entity type t, and the probability that the entity pair z is related with r, respectively. $P(t|e)$ is defined as follows:

$$P(t|e) = \sigma(\mathbf{W}_{\mathcal{T}_2} * \tanh(\mathbf{W}_{\mathcal{T}_1}\mathbf{X}) + \mathbf{b}_t), \tag{6}$$

where $\mathbf{X} \in \mathbb{R}^{2d}$ indicates the concatenation of a mention representation and context representations. Here, $\mathbf{W}_{\mathcal{T}_1}$ and $\mathbf{W}_{\mathcal{T}_2}$ are MLP parameter matrices, σ is the softmax classifier, and $\mathbf{b}_t$ is the bias.

Let $\mathbf{y}^e_j$ indicate the probability vector of the entity type predicted. The loss function $\mathcal{L}_{entity}$ is defined based on binary cross entropy as follows:

$$\mathcal{L}_{entity} = -\frac{1}{N}\sum_{j=1}^{N} \mathbf{x}^e_j \log \mathbf{y}^e_j + (1-\mathbf{x}^e_j)\log(1-\mathbf{y}^e_j), \tag{7}$$

where $\mathbf{x}^e_j$ indicates the binary vector where the value is 1 if it is a gold type.

Moreover, we can predict the relation type between two entities. The entity types, t_1 and t_2, predicted from e_1 and e_2 directly match to the complex-valued vectors $\mathbf{t}_1$ and $\mathbf{t}_2$. $P(r|z)$ is defined as follows:

$$P(r|z) = \sigma(\mathbf{W}_{\mathcal{R}} * s(t_1, t_2) + \mathbf{b}_r), \tag{8}$$

where $\mathbf{W}_{\mathcal{R}}$ is the parameter matrix, t_1 and t_2 are the gold truth entity types $\sigma()$ is the softmax function, and $\mathbf{b}_r$ is the bias.

The loss function for relation extraction $\mathcal{L}_{relation}$ and the final loss function $\mathcal{L}_{final}$ are defined as follows:

$$\begin{aligned}\mathcal{L}_{relation} = -\frac{1}{N^2}\sum_i\sum_j \mathbf{x}^r_{ij}\log \mathbf{y}^r_{ij} \\ + (1-\mathbf{x}^r_{ij})\log(1-\mathbf{y}^r_{ij})),\end{aligned} \tag{9}$$

$$\mathcal{L}_{final} = \mathcal{L}_{entity} + \lambda\mathcal{L}_{relation}, \tag{10}$$

where $\mathbf{x}^r_{ij}$ indicates the binary vector where the value is 1 if it is a gold type, and $\mathbf{y}^r_{ij}$ indicates the probability vector of relation type between t_i and t_j. λ is the weighting parameter.

Algorithm 1 Active Learning with a Granularity Score

Input: annotated training corpus $\mathcal{D}^l$, seed entity type hierarchy $\mathcal{T}$, entity typing and relation extraction model L
Output: expanded entity type hierarchy $\mathcal{T}_{expanded}$, updated training corpus $\mathcal{D}^l_{updated}$

```
train a neural model L with training corpus D^l, set δ* = 0
while δ* < threshold do
    compute the granularity score δ for each entity type appeared in D^l
    select t* ∈ T with lowest granularity score δ*
    select a mini-batch B of annotated sentences with t* from D^l
    D^u ← sentence including mentions annotated with t*
    D^l ← D^l \ D^u ∪ B
    retrain model L with D^l
    T ← T ∪ t^new
D^l_updated ← D^l, T_expanded ← T
Return T_expanded, D^l_updated
```

5 Active Learning

In this section, we introduce the active learning strategy to select the semantic types that need to be expanded. The workflow of our active learning model is shown in Algorithm 1. Our strategy computes the granularity of each type with the training corpus and determines whether we should add a new type to a KG schema. Since a relation type can be expanded through the same process, we only describe the expansion mechanism of an entity type in this section.

For each entity type $t \in \mathcal{T}$, we consider a granularity score as an uncertainty measure. We assume that if the granularity score of an entity type is low, the entity type needs to have a child or a sibling type. Based on the granularity scores, we select the type t^* with the lowest score δ^* among the entity types below a given threshold. The granularity score of entity type t_i is calculated as follows:

$$G(t_i) = -\frac{\sum_{j=1}^{k}\log(P(t_i|e_j))}{k}, \tag{11}$$

where e_j indicates an entity appeared in the given training corpus, and k is the number of all the entities.

Then, we query human users to create a new entity type t^{new} and annotate a mini-batch which contains the sentences annotated with t^*. After that, we add the newly annotated sentences to the training corpus and re-train the neural model with expanded schema.

However, it should be considered that the annotator might create a noisy type which decreases the accuracy of type prediction and the stability of a KG schema. For example, an entity *LionelMessi* might be annotated with a new entity type ⟨*person/sportsperson*⟩, even though an entity type ⟨*person/athlete*⟩ already exists. In this case, predicting the entity type of *LionelMessi* between the two types that represent nearly the same concept can be confusing. We detect this labeling noise by estimating the expected entropy of newly annotated entities. If the expected entropy is higher than the entropy before the expansion, the annotator is then requested to remove or relabel the corresponding type. The entropy is calculated as follows:

$$E = -\sum_i \sum_j P(t_j|e_i) * \log_2 P(t_j|e_i). \tag{12}$$

6 EXPERIMENTS

In order to evaluate the effectiveness of our framework, we used two publicly available corpora and two kinds KG schemas derived from Freebase. We compared the performance of the proposed framework on three sets of experiments with state-of-the-art baselines; 1) schema expansion; 2) entity typing and relation extraction; and 3) selection algorithms in active learning.

6.1 Datasets and Setup

6.1.1 *Corpora*

Both corpora, NYT and Wiki-KBP, include the sentences heuristically labeled with the Freebase schema via distant supervision [19]. (1) NYT: It consists of 236k annotated sentences sampled from New York Times news articles. For the test, 395 sentences were manually annotated by [37] [37] with 47 entity types and 24 relation types. (2) Wiki-KBP: The training data consists of 24k annotated sentences sampled from Wikipedia articles. 289 sentences were sampled from the 2013 KBP slot filling assessment results for the test data [38]. It is annotated with 126 entity types and 13 relation types. For both corpora, we used the training/validation/test set introduced in the prior work [19].

6.1.2 *Knowledge graph schema*

To evaluate the effectiveness of our framework, we constructed seed and target schemas for each corpus in advance. In the experiment, we considered the expansion of both entity types and relation types. The seed and target schemas for the entity type expansion have the same number of relation types. We initially designed domain-independent seed schemas, which consist of general concept as classes. (1) NYT Schema for entity type expansion: The seed schema has 17 entity types, and the target schema has 47 entity types; (2) Wiki-KBP Schema for entity type expansion: The seed schema has 76 entity types, and the target schema has 126 entity types.

Also, we constructed the seed and target schemas for relation type expansion. The seed and target schemas for the relation type expansion have the same number of entity types. (1) NYT Schema for relation type expansion: The seed schema has 4 relation types, and the target schema has 24 relation types; (2) Wiki-KBP Schema for relation type expansion: The seed schema has 8 relation types, and the target schema has 13 relation types.

6.1.3 *Evaluation metrics*

We mainly used the strict F1 score (Acc) to evaluate the performance of our proposed framework. Evaluating the performance of entity typing, we used additional metrics, including Macro-averaged F1 (Ma-F1) and Micro-averaged F1 (Mi-F1), which have been used for evaluating entity typing systems.

6.2 Baselines

To test the performance of schema expansion, we empirically compared our framework with UCOP, the approach proposed by [4]. A user manually defines entity types by organizing extracted entities from a corpus, and scouts for a target KG schema, which can fit the user data. After the target schema is selected, UCOP aligns user-defined types to the target KG schema by selecting a learning model and adds new types by computing a type probability entropy. In this experiment, a vanilla CNN based UCOP was implemented with the Stanford CoreNLP[3] tool as the entity extractor.

To evaluate the performance of entity typing of the proposed neural model, we compared our approach with the five state-of-the-art models: (1) Attentive [8]; (2) CoType [19]; (3) NFETC [12]; (4) KNET [39]; (5) Hierarchy [13]. To test the performance of relation extraction, we compared AKSE with the five relation extraction models: (1) FCM [40]; (2) MultiR [37]; (3) CoType [19]; (4) ReTrans [20]; (5) HRL [21].

We could justify the effectiveness of a granularity score as an uncertainty measure via experiments. Also, we compared the proposed granularity based active learning algorithm with three traditional selection strategies: random sampling (RAND); least confidence (LC); and margin sampling (MG) [41].

In addition, we designed the variants of our proposed model: (1) AKSE: the original framework; (2) AKSE-noatt: neural model without KG schema attention; (3) AKSE-noND: active learning without noisy type detection.

6.2.1 *Parameter settings*

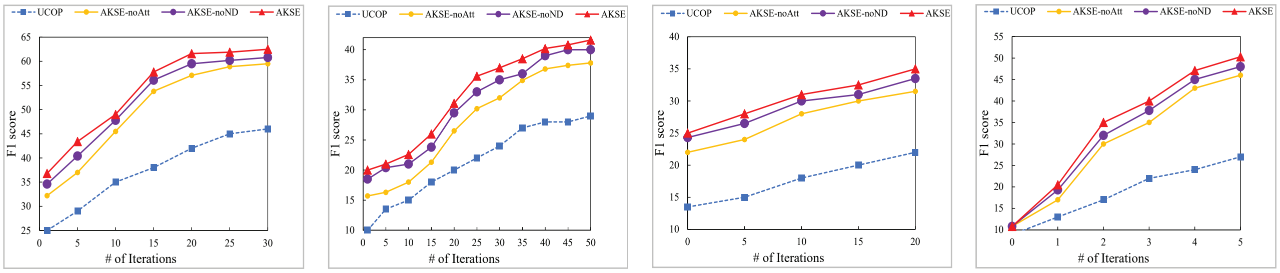
We used the pre-trained 300-dimensional word embeddings supplied by [42]. For the neural model, learning rate l_r and L2 regularization parameter λ were set to 0.001 and 0.0001 respectively, which were optimized by Adam Optimizer based on the validation set. In addition, we employed Dropout [43] on mention and context representations to avoid overfitting. We set the hyperparameters for KG schema embedding following [36]. For active learning, we set the threshold for Algorithm 1 as 1, and calculated the probability $P(t|e)$ to two decimal places. To use the optimal parameters for the baseline methods, we obtained the best values by analyzing the models' performance on a validation set that contains 10% of the data.

6.3 Performance

6.3.1 *Simulated schema expansion*

To evaluate our proposed methods, we conducted simulated schema expansion experiments on both datasets. For AKSE, we simulated manual annotation by using gold labels for the data selected by our active learning algorithm. On the other hand, in the case of UCOP, three users investigated the entities and relations extracted from the neural model and manually select the one that needs to be expanded for 5 minutes (average runtime of our active learning algorithm at each iteration). In order to compare under the same simulation environment, user-selected ones are annotated

3. https://stanfordnlp.github.io/CoreNLP/



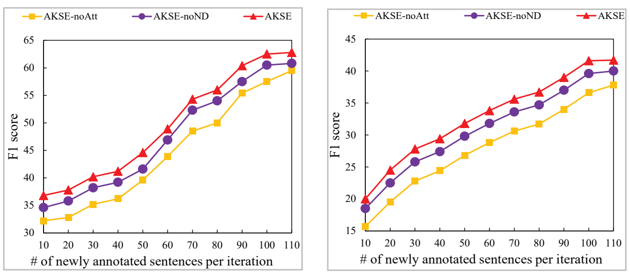
(a) performance comparison of entity type expansion on the NYT dataset

(b) performance comparison of entity type expansion on the Wiki-KBP dataset

(c) performance comparison of relation type expansion on the NYT dataset

(d) performance comparison of relation type expansion on the Wiki-KBP dataset

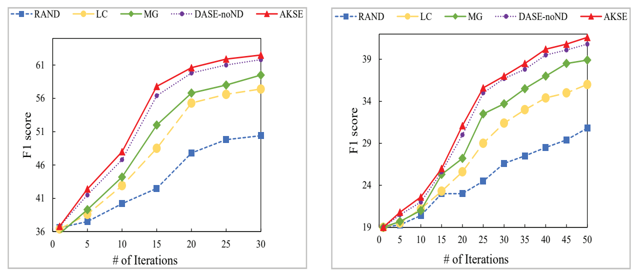
Fig. 4. Evaluation results on schema expansion



(a) impact of # of mini-batches on the NYT dataset

(b) impact of # of mini-batches on the Wiki-KBP dataset

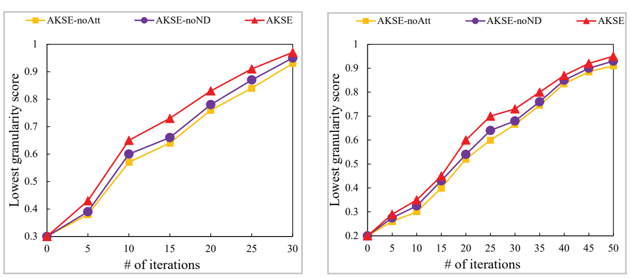
Fig. 5. Performance comparison of entity type expansion with a variation of mini-batch size



(a) performance comparison of active learning methods on the NYT dataset

(b) performance comparison of active learning methods on the Wiki-KBP dataset

Fig. 7. Evaluation results on active learning algorithm



(a) the lowest granularity score per iteration on the NYT dataset

(b) the lowest granularity score per iteration on the Wiki-KBP dataset

Fig. 6. The lowest granularity score per iteration

with the gold labels. For instance, when users determine that the entity 'Galleon Group' of '*organization*' type needs to be expanded, the name of an expanded type is set to the gold label '*organization/company*,' rather than a manually named like '*organization/enterprise*.'

At each iteration of the entity type expansion, all the methods added one entity(or relation) type to the schema and got the curated mini-batch and the expanded KG schema to re-train the models. The ultimate goal is to minimize the distance between $\mathcal{S}^{\Psi}_{expanded}$ and the target schema. We set the mini-batch size of an active learning process as 100 in this experiment. We ran experiments with five random initializations and report the average. Figure 4(a) and 4(b) show that our models outperform UCOP in entity type expansion on both datasets. Also, as shown in Figure 4(c) and 4(d), we could observe that our models perform much better than the baseline model in relation type expansion. At the end of the iteration of both entity type and relation type expansion, AKSE achieves about 20% higher F1 score than UCOP on NYT dataset, and 15% higher F1 score on Wiki-KBP dataset. Besides, we see that all the methods in entity type expansion perform better on the NYT dataset. Since NYT dataset has fewer entity types than the Wiki-KBP dataset, the expansion to the NYT target schema can be performed more accurately. Figure 4(c) and 4(d) show similar results in relation expansion, where all methods achieve better performance on Wiki-KBP dataset.

The results demonstrate that considering the granularity of types (AKSE) is more valid for schema expansion than considering the entropy of types (UCOP). Especially at the beginning of the iteration of the entity type expansion, UCOP starts about 7% behind the AKSE. The reason for the initial difference is that UCOP only uses an extracted entity mention embedding to classify the corresponding entity type, implying that it does not reflect the deep context of sentences.

We further compared the performance of variants of AKSE. We observed that AKSE achieves 5% higher F1 Score than AKSE-noatt, which indicates that the KG schema attention allows our model to predict more accurate entity types. We also found that noisy type detection process improves performance by about 2% through capturing the noisy type, which could be generated intermittently in the active learning phase.

To check the proposed method's model training efficiency empirically, we first initialized the models using 36% and 60% of training data on NYT and Wiki-KBP dataset, respectively, in the entity type expansion setting (16% and

TABLE 1
Average model training time per iteration in the simulation setting (in seconds).

Model	Entity type expansion		Relation type expansion	
	NYT	Wiki-KBP	NYT	Wiki-KBP
UCOP	11,350	3,420	9,200	3,050
AKSE	11,700	3,900	9,650	3,540

60% of training data on each dataset in the relation type expansion). We measure the average retraining time at each iteration where one entity (or relation) type is added to the schema. As reported in Table 1, AKSE shows similar training time to the compared method that uses vanilla CNN and simple entropy score as an uncertainty measure. We confirmed that the proposed method outperforms the compared method in schema expansion with a reasonable computational cost.

6.3.2 Human-in-the-loop schema expansion

We employed five annotators and let them annotate labels independently. A single annotator solely annotated batches during the full expansion phase for 3 minutes per training iteration. To compare the performance of schema expansion, we calculated the average F1 score of five annotators' results. Table 2 reports the results of the human annotation experiments. It shows that schema expansion with human-in-the-loop improved the performance by about 3.3% - 8.7% compared to the simulated schema expansion. We confirmed that the false positive expanded samples in the human-in-the-loop experiments are two times lower than those in the simulation. And once again, we could observe that the smaller the number of types included in the schema, the better the schema expansion.

From both the simulation and human-in-the-loop experiments, we can confirm that our approach is successful in assigning new types detected in corpora not relying on domain expertise.

6.3.3 Impact of batch size

We explored the impact of the number of newly annotated sentences in the schema expansion, especially in the entity type expansion. We iterated a learning phase 30 times for each dataset. As shown in Figure 5(a) and 5(b), F1 scores rise on both datasets until the mini-batch size reaches 100, but the performance hardly improves after that. We note that if the size of the mini batches exceeds a certain number, there is little performance improvement (about 0.3%) due to the data redundancy.

A human user only updates the small number of sentences in the mini-batch while the corpus has tens of thousands of sentences. We needed to refine the imbalanced corpus before re-training the neural model where only a few sentences are annotated with the new type. Thus, we found remaining sentences need to be annotated with the new type by estimating the cosine similarity of sentence representations between sentences in the mini-batch and those not in the mini-batch. Then, we newly annotated sentences above the similarity threshold 0.9.

TABLE 2
Evaluation results of schema expansion with human annotator. We measure strict F1 scores when the seed schemas extended to the full target schemas

Model	Entity type expansion		Relation type expansion	
	NYT	Wiki-KBP	NYT	Wiki-KBP
AKSE(simulated)	62.5	41.6	35	43.7
AKSE(human)	65.8	47.0	50.3	59

TABLE 3
Performance of comparison entity typing (%).

Model	NYT			Wiki-KBP		
	Acc	Ma-F1	Mi-F1	Acc	Ma-F1	Mi-F1
Attentive	57.4	63.8	63.4	35.7	57.7	57.8
CoType	59.7	64.3	63.8	38.5	60.9	57.3
NFETC	61.5	65.4	64.9	39.1	62.3	58.8
Hierarchy	60.8	65.2	65.0	39.7	62.8	59.3
KNET	61.2	64.6	65.3	39.8	60.9	61.0
AKSE (Bilinear)	61.8	66.0	65.5	41.2	63.4	62.0
AKSE (ComplEx)	**62.2**	**66.5**	**66.0**	**42.6**	**64.5**	**62.8**

6.3.4 Active Learning

We evaluated the validity of the granularity scores as an uncertainty measure. As shown in Figure 6(a) and 6(b), the lowest granularity score per iteration is increasing steeply. It is important to note that the growing trend of granularity scores justifies that granularity score is a suitable and valid measure for representing the level of completeness of the KG schema.

We also evaluated the performance of a selection strategy in active learning. During the evaluation, the active learning phase was conducted by human evaluators. In Figure 7(a) and 7(b), RAND shows the worst performance among the compared algorithms. MG and GRAN outperform LC, which indicates that traditional LC does not identify the potential type that needs expansion. The proposed algorithm shows 4% higher F1 Score than MG. The reason is that our active learning algorithm computes a granularity of the type hierarchy with the hypothesis that a coarse type should be organized in a set of specific types. Meanwhile, MG ignores much of the type distribution with a large number of type sets. Furthermore, we could observe that AKSE performs slightly better than AKSE-noND with about 1% higher score. It shows that noisy types can be detected in active learning phase and relabeling increases the expansion accuracy.

6.3.5 Entity Typing

Table 3 compares our methods with state-of-the-art entity typing baselines on NYT and Wiki-KBP datasets. It shows that our model performs better than state-of-the-art methods on both datasets. In particular, our model is much better on Wiki-KBP dataset (about 3% higher under strict accuracy metric) while slightly ahead on NYT dataset (about 1% higher under strict accuracy metric). It is worth noting that the more the number of entity types is, the stronger the influence of incorporating the KG schema attention becomes.

We also analyzed the role of KG schema encoder for entity typing. We compared our complete model using

TABLE 4
Performance comparison of relation extraction. (%).

Model	NYT			Wiki-KBP		
	Prec	Rec	F1	Prec	Rec	F1
FCM	35.3	28.0	31.2	27.6	19.7	22.9
MultiR	29.3	30.5	29.9	25.0	20.7	22.6
CoType	38.8	47.0	42.5	30.7	37.1	33.6
ReTrans	41.3	48.8	44.7	28.9	41.2	34.0
HRL	48.9	48.5	48.6	**31.5**	39.8	35.1
AKSE (Bilinear)	48.8	48.9	48.8	30.5	41.1	35.0
AKSE (ComplEx)	**49.0**	**48.9**	**48.9**	31.0	**41.5**	**35.4**

ComplEx embedding method for KG schema encoding, i.e., AKSE (ComplEx) with the variant model using bilinear embedding method for KG schema encoding, i.e., AKSE (Bilinear). The bilinear model is equivalent to RESCAL [44] with a single is_A relation type, which does not consider the antisymmetric relations of entity type hierarchies. AKSE (ComplEx) model outperforms the model using the bilinear embedding model, implying that it is crucial in dealing with hierarchical schema information for entity typing.

Furthermore, we analyzed the common causes of entity typing errors, coupled with typical examples:

- Data imbalance: About 80% of the entity mentions have the '*location*' type in training data on NYT dataset. While our model predicted the '*location*' type with an accuracy of over 75%, errors frequently occured on other types of entities. In Wiki-KBP, about 20% of the entity mentions have the '*location/city*' type, and 67% of the entity mentions are one of the sub-types of '*person*,' e.g. '*person/soldier*'. It means that since the distribution of training data in the Wiki-KBP dataset is well-balanced compared to that in NYT dataset, the performance improvement in Wiki-KBP was much greater than that in NYT dataset.
- Ambiguous types: In the NYT dataset, given a sentence "...director of the advanced residency program in photograph conservation at the George Eastman House in Rochester," our model predicted '*location*' while the gold type is '*organization*.' Also, in the Wiki-KBP dataset, our model output '*/person* ' when the gold type is '*/organization*' for "Philips will replace outgoing Morrison chief Marc Bolland, who is leaving to take the top job at British clothes-to-food giant Marks and Spencer". The error of ambiguous types took up about 25% of all errors.
- Unnecessary sub-types: Sometimes, our model predicted unnecessary sub-types. In this type of error, our model correctly predicted the upper type, but unnecessarily predicted the wrong sub-type. For example, in the Wiki-KBP dataset, given a sentence "Hwang Jang-Yop, a former secretary of North Korea's ruling Workers Party, is credited with ...", while the gold type is '*government/political_party*' and '*organization*' , but our model predicted '*organization/company*'. This error took up about 18% of all errors.

6.3.6 Relation Extraction

Table 4 compares our methods with state-of-the-art relation extraction baselines on NYT and Wiki-KBP datasets. It shows that our model achieves competitive performance among state-of-the-art relation extraction models on both datasets. Our model is comparable to HRL and performs better than other baselines on the Wiki-KBP dataset. Although CoType considers various lexical features of the text itself and ReTrans transfers knowledge from other domains to a target domain, the results indicate that the correlation between entity types and relation types in a given KG schema plays an essential role in predicting a relation type between entities.

Like the entity typing experiment, we studied the impact of hierarchical schema attention on the AKSE (Bilinear) model. Our complete model shows a slightly better performance than the simpler model. We can verify that KG schema attention also plays an important role in relation extraction.

We also identified the common causes of relation extraction errors coupled with typical examples:

- Confusing sub-types: As shown in Table 4, the results demonstrate that our model's performance is only slightly better than other methods on the NYT dataset. We found that about 72% of relation types are the sub-types of '*location*' such as '*location/location/contains*' or '*location/country/capital*.' Our model failed to strictly distinguish the sub-types of '*location*' in the testing phase.
- Inverse types: In the Wiki-KBP dataset, our model output '*org:subsidiaries*' type when the gold type is '*org:parents*' for "German drug and chemical maker Merck KGaA says third-quarter earnings rose 46 percent, as it reported revenues from new acquisition Millipore Corp...".
- Semantically similar types: Our model occasionally confused several semantically similar relation types. For instance, in Wiki-KBP dataset, our model outputs '*per:country_of_death*' while the gold type is '*per:country_of_residence*' in "Kissel, a native of Adrian, Michigan, whose family has also lived in Minneapolis, has been serving a life sentence since she was convicted in September 2005." It seems that our model focused on the context of "life sentence."
- Long distance: When the distance between entities becomes longer, our model failed to predict the gold type. For example, our model outputs '*location/contains*' whereas the gold type is '*location/administrative_division/country*' for the NYT instance "...who draw on Syria and Saudi Arabia for money and other logistical support , was orders of magnitude greater than that from Shiites , and he contended that the Bush administration 's public emphasis on the E.F.P. 's was part of a larger administration strategy to blame Iran...". We have a plan to elaborate the context representation as a future work.

7 Conclusion

We introduced the active learning framework, which can expand a KG schema from an initial KG schema. We designed the neural model for entity typing and relation extraction that leverages type information of KG schemas. We also presented an active learning algorithm, which selects the types that need expansion, considering the granularity of types. Experimental results on two publicly available datasets demonstrate the effectiveness and validity of the proposed approach. As future work, we will study effective re-training schemes for the schema expansion method. We plan to address complex relations such as n-ary relations, which are expressed in natural languages more frequently.

ACKNOWLEDGEMENT

This work was supported by NCSOFT NLP Center.

REFERENCES

[1] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, "Deep learning based text classification: A comprehensive review," *arXiv preprint arXiv:2004.03705*, 2020.
[2] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu, "A survey on knowledge graphs: Representation, acquisition and applications," *arXiv preprint arXiv:2002.00388*, 2020.
[3] M. Mintz, S. Bills, R. Snow, and D. Jurafsky, "Distant supervision for relation extraction without labeled data," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*. Association for Computational Linguistics, 2009, pp. 1003–1011.
[4] K. Clarkson, A. L. Gentile, D. Gruhl, P. Ristoski, J. Terdiman, and S. Welch, "User-centric ontology population," in *European Semantic Web Conference*. Springer, 2018, pp. 112–127.
[5] D. Yogatama, D. Gillick, and N. Lazic, "Embedding methods for fine grained entity type classification," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, vol. 2, 2015, pp. 291–296.
[6] L. Dong, F. Wei, H. Sun, M. Zhou, and K. Xu, "A hybrid neural model for type classification of entity mentions." in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015, pp. 1243–1249.
[7] S. Shimaoka, P. Stenetorp, K. Inui, and S. Riedel, "An attentive neural architecture for fine-grained entity type classification," in *Proceedings of the 5th Workshop on Automated Knowledge Base Construction*, 2016, pp. 69–74.
[8] ——, "Neural architectures for fine-grained entity type classification," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, 2017, pp. 1271–1280.
[9] X. Ren, W. He, M. Qu, L. Huang, H. Ji, and J. Han, "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, pp. 1369–1378.
[10] X. Ren, W. He, M. Qu, C. R. Voss, H. Ji, and J. Han, "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*. ACM, 2016, pp. 1825–1834.
[11] A. Abhishek, A. Anand, and A. Awekar, "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: Volume 1, Long Papers*, vol. 1, 2017, pp. 797–807.
[12] P. Xu and D. Barbosa, "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)*, vol. 1, 2018, pp. 16–25.
[13] S. Murty, P. Verga, L. Vilnis, I. Radovanovic, and A. McCallum, "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)*, vol. 1, 2018, pp. 97–109.
[14] X. Zhong, E. Cambria, and A. Hussain, "Extracting time expressions and named entities with constituent-based tagging schemes," *Cognitive Computation*, pp. 1–19, 2020.
[15] Y. Ma, E. Cambria, and S. Gao, "Label embedding for zero-shot fine-grained named entity typing," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 2016, pp. 171–180.
[16] L. Huang, J. May, X. Pan, and H. Ji, "Building a fine-grained entity typing system overnight for a new x (x= language, domain, genre)," *arXiv preprint arXiv:1603.03112*.
[17] I. Augenstein, A. Vlachos, and D. Maynard, "Extracting relations between non-standard entities using distant supervision and imitation learning," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 747–757.
[18] Y. Yaghoobzadeh, H. Adel, and H. Schütze, "Noise mitigation for neural entity typing and relation extraction," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, 2017, pp. 1183–1194.
[19] X. Ren, Z. Wu, W. He, M. Qu, C. R. Voss, H. Ji, T. F. Abdelzaher, and J. Han, "Cotype: Joint extraction of typed entities and relations with knowledge bases," in *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2017, pp. 1015–1024.
[20] S. Di, Y. Shen, and L. Chen, "Relation extraction via domain-aware transfer learning," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2019, pp. 1348–1357.
[21] R. Takanobu, T. Zhang, J. Liu, and M. Huang, "A hierarchical framework for relation extraction with reinforcement learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 7072–7079.
[22] X. Zhang, P. Li, W. Jia, and H. Zhao, "Multi-labeled relation extraction with attentive capsule network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 7484–7491.
[23] G. Singh and P. Bhatia, "Relation extraction using explicit context conditioning," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2019, pp. 1442–1447.
[24] M. Kholghi, L. Sitbon, G. Zuccon, and A. Nguyen, "Active learning reduces annotation time for clinical concept extraction," *International journal of medical informatics*, vol. 106, pp. 25–31, 2017.
[25] H. Al-Olimat, S. Gustafson, J. Mackay, K. Thirunarayan, and A. Sheth, "A practical incremental learning framework for sparse entity extraction," in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 700–710.
[26] C. Wang, L. Chiticariu, and Y. Li, "Active learning for black-box semantic role labeling with neural factors," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. AAAI Press, 2017, pp. 2908–2914.
[27] Y. Shen, H. Yun, Z. C. Lipton, Y. Kronrod, and A. Anandkumar, "Deep active learning for named entity recognition," in *Proceedings of the Sixth International Conference on Learning Representations*, 2018.
[28] Y. Zhang, M. Lease, and B. C. Wallace, "Active discriminative text representation learning." 2017, pp. 3386–3392.
[29] S. Peshterliev, J. Kearney, A. Jagannatha, I. Kiss, and S. Matsoukas, "Active learning for new domains in natural language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Industry Papers)*, 2019, pp. 90–96.
[30] P. Hu, Z. C. Lipton, A. Anandkumar, and D. Ramanan, "Active learning with partial feedback," in *Proceedings of the 7th international conference on learning representations*, 2019, pp. 1–15.
[31] A. Chaudhary, J. Xie, Z. Sheikh, G. Neubig, and J. G. Carbonell, "A little annotation does a lot of good: A study in bootstrapping low-resource named entity recognizers," 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)*, 2019, pp. 5167–5177.
[32] J. Kasai, K. Qian, S. Gurajada, Y. Li, and L. Popa, "Low-resource deep entity resolution with transfer and active learning," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 5851–5861.

[33] B. Z. Li, G. Stanovsky, and L. Zettlemoyer, "Active learning for coreference resolution using discrete annotation," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2020, pp. 8320–8331.

[34] H. Paulheim, "Knowledge graph refinement: A survey of approaches and evaluation methods," *Semantic web*, vol. 8, no. 3, pp. 489–508, 2017.

[35] W. Yin, H. Schütze, B. Xiang, and B. Zhou, "Abcnn: Attention-based convolutional neural network for modeling sentence pairs," *Transactions of the Association of Computational Linguistics*, vol. 4, no. 1, pp. 259–272, 2016.

[36] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," in *International Conference on Machine Learning*, 2016, pp. 2071–2080.

[37] R. Hoffmann, C. Zhang, X. Ling, L. Zettlemoyer, and D. S. Weld, "Knowledge-based weak supervision for information extraction of overlapping relations," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, 2011, pp. 541–550.

[38] J. Ellis, X. Li, K. Griffitt, S. Strassel, and J. Wright, "Linguistic resources for 2013 knowledge base population evaluations." in *TAC*, 2012.

[39] J. Xin, Y. Lin, Z. Liu, and M. Sun, "Improving neural fine-grained entity typing with knowledge attention," pp. 5997–6004, 2018.

[40] M. R. Gormley, M. Yu, and M. Dredze, "Improved relation extraction with feature-rich compositional embedding models," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1774–1784.

[41] Y. Fu, X. Zhu, and B. Li, "A survey on instance selection for active learning," *Knowledge and information systems*, vol. 35, no. 2, pp. 249–283, 2013.

[42] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[43] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[44] M. Nickel, V. Tresp, and H.-P. Kriegel, "A three-way model for collective learning on multi-relational data." in *Proceedings of the 28th International Conference on International Conference on Machine Learning*, vol. 11, 2011, pp. 809–816.

Seungmin Seo received his B.S. degree in Computer Science from Yonsei University, Seoul, Korea, in 2015. He is currently working toward a Ph.D. degree in the Department of Computer Science at Yonsei University. His research interests include knowledge representation, information extraction, and natural language processing.

Byungkook Oh received his B.S., M.S. and Ph.D degree in Computer Science from Yonsei University, Seoul, Korea, in 2015, 2017, and 2020. He is currently working as a researcher in Samsung Research. His research interests include knowledge graph embedding, natural language processing, and knowledge-based systems.

Eunju Jo received the B.S degree in Software Engineering from Dongseo University, Busan, South Korea, in 2017. She received the M.S degree in the Department of Computer Science at Yonsei University, Seoul, Korea, in 2019. She is currently working as a researcher in KT. Her research interests include information retrieval and natural launguage processing.

Sanghak Lee received his B.S. and M.S. degree in Computer Science from Yonsei University, Seoul, Korea, in 2018 and 2020, respectively. He is currently working as a researcher in TmaxAI. His research interests include knowledge graph and natural language processing.

Dongho Lee received his B.S. degree in Computer Science from Brandeis University, MA, USA, in 2018. He received the M.S degree in the Department of Computer Science at Yonsei University, Seoul, Korea, in 2020. He is currently working as a researcher in LG Electronics. His research interests include knowledge graph, recommender system, and natural language processing.

Kyong-Ho Lee is currently a professor in the Department of Computer Science at Yonsei University. He received his B.S., M.S. and Ph.D. degrees in Computer Science from Yonsei University, Seoul, South Korea, in 1995, 1997, and 2001, respectively. Previously, he worked as a guest researcher in the IT Laboratories at NIST (National Institute of Standards and Technology), Maryland. He was a visiting scholar at University of California, Irvine, in 2008. His research interests include Knowledge Graph and Semantic Web.

Dong-hoon Shin is currently a senior AI researcher at NCSOFT. He received his B.S., M.S. and Ph.D. degrees in Computer Science from Yonsei University, Seoul, South Korea. Previously, he was a visiting researcher at NIST (National Institute of Informatics), Tokyo, Japan. He worked as an AI researcher at CTO of LG Electronics. His research interests includes intelligent agent, Q&A over knowledge base and natural language processing.

Yeonsoo Lee is Executive Director at NCSOFT Co Ltd. She has been leading the Language AI Lab for five years. She received his B.S., M.S. and Ph.D. degrees in Computer Science from Korea University, Seoul, South Korea, in 2000, 2008, and 2014, respectively. She worked as a software engineer at Samsung SDS Co Ltd. from 2000 to 2004. Her research interests include dialogue system, machine translation, and information extraction.