Open Hierarchical Relation Extraction

Kai Zhang[1*], **Yuan Yao**[1*], **Ruobing Xie**[2],
Xu Han[1], **Zhiyuan Liu**[1†], **Fen Lin**[2], **Leyu Lin**[2], **Maosong Sun**[1]
[1]Department of Computer Science and Technology
Institute for Artificial Intelligence, Tsinghua University, Beijing, China
Beijing National Research Center for Information Science and Technology, China
[2]WeChat Search Application Department, Tencent, China
drogozhang@gmail.com, yaoyuanthu@163.com

Abstract

Open relation extraction (OpenRE) aims to extract novel relation types from open-domain corpora, which plays an important role in completing the relation schemes of knowledge bases (KBs). Most OpenRE methods cast different relation types in isolation without considering their hierarchical dependency. We argue that OpenRE is inherently in close connection with relation hierarchies. To address the bidirectional connections between OpenRE and relation hierarchy, we propose the task of open hierarchical relation extraction and present a novel OHRE framework for the task. To effectively integrate hierarchy information into relation representations for better novel relation extraction, we propose a dynamic hierarchical triplet objective and hierarchical curriculum training paradigm. We also present a top-down hierarchy expansion algorithm to add the extracted relations into existing hierarchies with reasonable interpretability. Comprehensive experiments show that OHRE outperforms state-of-the-art models by a large margin on both relation clustering and hierarchy expansion. The source code and experiment details of this paper can be obtained from https://github.com/thunlp/OHRE.

1 Introduction

Open relation extraction (OpenRE) aims to extract novel relations types between entities from open-domain corpora, which plays an important role in completing the relation schemes of knowledge bases (KBs). OpenRE models are mainly categorized into two groups, namely tagging-based and clustering-based methods. Tagging-based methods consider OpenRE as a sequence labeling task, which extracts relational phrases from sentences (Banko et al., 2007; Cui et al., 2018). In contrast, clustering-based methods aim to cluster relation instances into groups based on their semantic similarities, and regard each cluster as a relation (Yao et al., 2011; Wu et al., 2019).

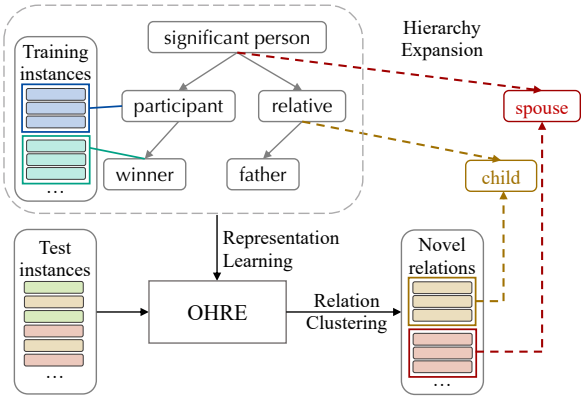


Figure 1: The workflow of OHRE framework. Trained with relation hierarchy and labeled instances, OHRE extracts novel relations from open-domain corpora and adds them into the existing hierarchy.

However, most OpenRE models cast different relation types in isolation, without considering their rich hierarchical dependencies. Hierarchical organization of relations has been shown to play a central role in the abstraction and generalization ability of human (Tenenbaum et al., 2011). This hierarchical organization of relations also constitutes the foundation of most modern KBs (Auer et al., 2007; Bollacker et al., 2008). Figure 1 illustrates an example of relation hierarchy in Wikidata (Vrandečić and Krötzsch, 2014). Such relation hierarchies are crucial in establishing the relation schemes of KBs, and could also help users better understand and utilize relations in various downstream tasks.

However, manually establishing and maintaining the ever-growing relation hierarchies require

* indicates equal contribution
† Corresponding author: Z.Liu (liuzy@tsinghua.edu.cn)

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 5682–5693
June 6–11, 2021.

expert knowledge and are time-consuming, given the usually large quantity of relations in existing hierarchy and the rapid emergence of novel relations in open domain corpora.[1] Since the ultimate goal of OpenRE is to automatically establish and maintain relation schemes for KBs, it is desirable to develop OpenRE methods that can directly add the extracted novel relations into the existing incomplete relation hierarchy. Moreover, incorporating the hierarchical information of existing relations can also help OpenRE methods to model their interdependencies. Such refined semantic connections among existing relations can provide transferable guidance to better extract new relations.

Given the inherent bidirectional connections between OpenRE and relation hierarchy, in this work, we aim to introduce relation hierarchy information to improve OpenRE performance, and directly add the extracted new relations into the existing hierarchy, which presents unique challenges. We propose a novel framework **OHRE** to consider relation hierarchy in OpenRE. The key intuition behind our framework is that distance between relations in hierarchy reflects their semantic similarity. Therefore, nearby relations should share similar representations, and vice versa. Figure 1 shows the framework of OHRE, which consists of two components:

(1) In *relation representation learning*, we design a dynamic hierarchical triplet objective to integrate hierarchy information into relation representations. We also present a hierarchical curriculum learning strategy for progressive and robust training. (2) In *relation hierarchy expansion*, we first cluster instances into new relation prototypes and then conduct a top-down hierarchy expansion algorithm to locate new relations into hierarchy. In this way, OHRE encodes hierarchical information into relation representations, which improves classical OpenRE and further enables hierarchy expansion.

To verify the effectiveness of hierarchical information and the proposed framework, we conduct experiments over two evaluations, including the classical relation clustering task and a novel hierarchy expansion task. Experimental results on two real-world datasets show that our framework can bring significant improvements on the two tasks, even with partially available hierarchy from KBs.

The main contributions of this work are concluded as follows: (1) To the best of our knowledge, we are the first to address bidirectional connections between OpenRE and relation hierarchy. We propose a novel open hierarchical relation extraction task, which aims to provide new relations and their hierarchical structures simultaneously. (2) We present a novel OHRE framework for the proposed task, which integrates hierarchical information into relation representations for better relation clustering, and directly expands existing relation hierarchies with a top-down algorithm. (3) Comprehensive experiments on two real-world datasets demonstrate the effectiveness of OHRE on both relation clustering and hierarchy expansion.

2 Related Works

Open Relation Extraction. Recent years have witnessed an upsurge of interest in open relation extraction (OpenRE) that aims to identify new relations in unsupervised data. Existing OpenRE methods can be divided into tagging-based methods and clustering-based methods. Tagging-based methods seek to extract surface form of relational phrases from text in unsupervised (Banko et al., 2007; Banko and Etzioni, 2008), or supervised paradigms (Angeli et al., 2015; Cui et al., 2018; Stanovsky et al., 2018). However, many relations cannot be explicitly represented as surface forms, and it is hard to align different relational tokens with the same meanings.

In contrast, traditional clustering-based OpenRE methods extract rich features of sentences and cluster features into novel relation types (Lin and Pantel, 2001; Yao et al., 2011, 2012; Elsahar et al., 2017). Marcheggiani and Titov (2016) propose discrete-state variational autoencoder (VAE) that optimizes a relation classifier by reconstruction signals. Simon et al. (2019) introduce skewness loss to enable stable training of VAE. Hu et al. (2020) learn relation representations and clusters iteratively via self-training. Wu et al. (2019) improve conventional unsupervised clustering-based methods by combining supervised and unsupervised data via siamese networks, and achieve state-of-the-art performance. However, existing OpenRE methods cast different relation types in isolation without considering their rich hierarchical dependencies.

Hierarchy Information Exploitation. Well-organized taxonomy and hierarchies can facilitate many downstream tasks. Hierarchical information derived from concept ontologies can reveal semantic similarity (Leacock and Chodorow, 1998;

[1] E.g., the number of relations in Wikidata has grown to more than $8,000$ in the last 6 years.

Ponzetto and Strube, 2007), and is widely applied in enhancing classification models (Rousu et al., 2005; Weinberger and Chapelle, 2009) and knowledge representation learning models (Hu et al., 2015; Xie et al., 2016). Similar to concept hierarchy, some recent works try to exploit semantic connections from relation hierarchy. In the field of relation extraction, Han et al. (2018a) propose a hierarchical attention scheme to alleviate the noise in distant supervision. Zhang et al. (2019) leverage implicit hierarchical knowledge from KBs and propose coarse-to-fine grained attention for long-tail relations. However, these methods are designed to identify pre-defined relations, and cannot be applied to OpenRE that aims to discover novel relations in open-domain corpora.

3 OHRE Framework

We divide the open hierarchical relation extraction problem into two phases: (1) learning relation representations with hierarchical information and (2) clustering and linking novel relations to existing hierarchies.

3.1 Relation Representation Learning

Learning relation representation is fundamental to open hierarchical relation extraction. We encode sentences into relation representations using a relation embedding encoder. We assume existing relations are organized in hierarchies, which is common in most modern KBs. Note that while Figure 1 shows one hierarchy tree, the relation hierarchies may contain multiple trees. To fully utilize hierarchy information, we design a dynamic hierarchical triplet objective that integrates hierarchy information into relation representations, and hierarchical curriculum learning for robust model training. Pairwise virtual adversarial training is also introduced to improve the representation generalization ability.

Relation Embedding Encoder. We adopt CNN to encode sentences into relation representations. Following previous works (Zeng et al., 2014), given a sentence s and target entity pair (e_h, e_t), each word in the sentence is first transformed into input representations by the concatenation of word embedding and position embedding indicating the position of each entity. Then the input representation is fed into a convolutional layer followed by a max-pooling layer and a fully-connected layer to obtain the relation representation $\boldsymbol{v} \in \mathbb{R}^d$. The relation representation is normalized by L2 norm, i.e., $\|\boldsymbol{v}\|_2 = 1$. The relation encoder can be denoted as:

$$\boldsymbol{v} = \mathrm{CNN}(s, e_h, e_t). \tag{1}$$

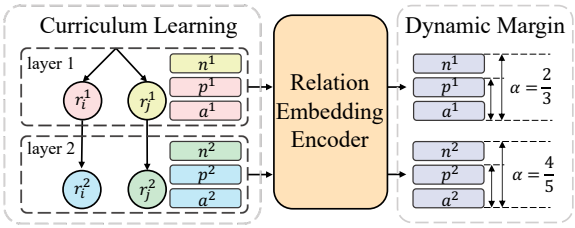


Figure 2: OHRE samples triplets from relations in hierarchy following a shallow-to-deep paradigm and sets dynamic margin via relation distance in hierarchy.

After obtaining relation representations, we measure the similarity of two relation instances by the Euclidean distance between their representations:

$$d(\boldsymbol{v}_1, \boldsymbol{v}_2) = \|\boldsymbol{v}_1 - \boldsymbol{v}_2\|_2^2. \tag{2}$$

Dynamic Hierarchical Triplet Loss. To effectively integrate relation hierarchy information into relation representations, we propose a dynamic hierarchical triplet loss for instance representation learning. Triplet loss is widely used in metric learning that encourages a static margin between different categories for distinguishment (Schroff et al., 2015). We argue that good relation representations should also reflect hierarchical information, where relations with close semantics in hierarchy should share similar representations. As the example shown in Figure 2, r_i^1 and r_j^1 should be closer than r_i^2 and r_j^2 in representation space, since r_i^1 and r_j^1 are close to each other in the relation hierarchy.

We design a hierarchical triplet objective with a dynamic margin which is determined by the distance between relations in hierarchy. Specifically, the dynamic margin is conducted over the instances of the relations. As shown in Figure 2, given two relations r_i and r_j sampled by hierarchical curriculum training strategy (which will be introduced later), we randomly sample two instances (namely anchor instance a and positive instance p) from r_i, and an instance (namely negative instance n) from r_j. The hierarchical triplet objective requires model to distinguish the positive pair (a, p) from the negative pair (a, n) by a distance margin, which is dynamically determined by the length of the shortest

path between r_i and r_j in the hierarchy as follows:

$$\mathcal{L}_t = \sum_{r_i, r_j \sim T} \max[0, d(\boldsymbol{v}_a, \boldsymbol{v}_p) + \lambda_d \frac{l(r_i, r_j)}{1 + l(r_i, r_j)} - d(\boldsymbol{v}_a, \boldsymbol{v}_n)], \tag{3}$$

where λ_d is a hyperparameter, $l(r_i, r_j)$ is the length of the shortest path between r_i and r_j in the hierarchy,[2] and T is the curriculum training strategy that will be introduced later. Intuitively, the margin increases with the length of the shortest path in the hierarchy, with a relative emphasis on distinguishing nearby relations. Compared to the static margin in vanilla triplet loss, dynamic hierarchical margin can capture the semantic similarities of relations in the hierarchy, leading to representations that can serve not only novel relation clustering but also effective relation hierarchy expansion.

Hierarchical Curriculum Learning. In addition to providing direct supervision for representation learning, relation hierarchy can also be useful in providing signals for robust model training. We propose a hierarchical training paradigm, which is a curriculum learning strategy (Bengio et al., 2009) that enables progressive training. The motivation is intuitive: In the early period of training, we choose relations that are easy to distinguish by the model, and gradually transfer to harder ones. Specifically, we sample two relations from the same layer in hierarchy that share ancestor relations (i.e., the relations come from the same tree and are of the same depth), with a gradual transition from shallow to deep layers with respect to their common ancestor, as shown in Figure 2. The training procedure will lead the model to learn relations from coarse to fine grains, since the length of the shortest path between two relations in hierarchy gradually increases as the relation pair goes deeper.[3] In experiments, we find it beneficial to warm-up the training of OHRE under the hierarchical training paradigm, and then switch to two random relations in the later phase.

Pair-wise Virtual Adversarial Training. Neural metric learning models may suffer from the overfitting problem by learning very complex decision hyperplanes. In our case, the problem is severe since relation hierarchies provide strong supervision to metric learning. To address this issue, we design pair-wise virtual adversarial training that smooths the representation space by penalizing sharp changes in the space. Specifically, for each randomly sampled instance pair, we add worst-case perturbations, such that the distance between the relation pairs reaches the maximum changes. We penalize the loss changes as follows:

$$\mathcal{L}_v = \sum_{v_1, v_2} \|d(\boldsymbol{v}_1, \boldsymbol{v}_2) - d(\tilde{\boldsymbol{v}}_1, \tilde{\boldsymbol{v}}_2)\|_2^2, \tag{4}$$

where $\tilde{\boldsymbol{v}}$ is obtained by adding the worst-case noise to $\boldsymbol{v}$. Pair-wise virtual adversarial training encourages smooth and robust metric space, thus improving the generalization ability of OpenRE models. Unlike previous works that adopt virtual adversarial training in classification problems (Miyato et al., 2017; Wu et al., 2019), our pair-wise virtual adversarial training is based on distance in Euclidean space instead of classification probability distributions. We refer readers to the appendix for more details about the pair-wise virtual adversarial training. The final loss is defined as the addition of dynamic hierarchical triplet loss $\mathcal{L}_t$ and pair-wise virtual adversarial loss $\mathcal{L}_v$:

$$\mathcal{L} = \mathcal{L}_t + \lambda_v \mathcal{L}_v, \tag{5}$$

where λ_v is a hyperparameter.

3.2 Relation Hierarchy Expansion

To expand the existing relation hierarchies, we first cluster novel relations in open-domain corpora based on instance representations, and then learn relation prototypes for both relations in the existing hierarchy and novel relations. Finally, new relations are inserted into the existing relation hierarchy by a novel top-down hierarchy expansion algorithm based on relation prototypes.

The hierarchy expansion framework is designed based on two key assumptions: (1) A relation prototype is the aggregation of all instances belonging to itself and descendant relations. (2) A relation prototype has the highest similarity with its parent relation prototype, and a lower similarity with its sibling relation prototypes. The rationale of the assumptions is that the semantics of a relation is typically covered by its ancestors. The assumption is also aligned with the intuition in relation representation learning, where a relation exhibits the highest similarity with its parent, due to the minimum shortest path length (i.e., the length is 1).

Relation Prototype Learning. We first cluster new relations in unsupervised data by Louvain algo-

[2] The margin is 1 if two relations come from different trees.

[3] Relations with longer shortest paths are more difficult to the model, since they need to be distinguished by larger margins, as indicated in Equation 3.

rithm (Blondel et al., 2008). Louvain detects communities in a graph by greedily merging data points to clusters based on modularity optimization, and has proven effective in OpenRE (Wu et al., 2019). We construct a weighted undirected graph of the relation instances in the test set, where the connection weight between two instances is determined by the distance between their representations:

$$w(\boldsymbol{v}_1, \boldsymbol{v}_2) = \max[0, 1 - d(\boldsymbol{v}_1, \boldsymbol{v}_2))]. \quad (6)$$

In experiments, we observe that clusters containing very few instances are typically noisy outliers and are not proper to be regarded as novel relations, which is consistent with Wu et al. (2019). Therefore, we merge instances in these clusters into their closest clusters, measured by the highest connection weight. Then we learn relation prototypes for both relations in the existing hierarchy and novel relations based on the clusters. We represent each relation prototype with instances, where the prototype of a novel relation consists of all its instances, and the prototype of an existing relation contains all instances from itself and all descendant relations.

Top-Down Hierarchy Expansion. After obtaining relation prototypes, we link these extracted relations to existing hierarchy by a novel top-down hierarchy expansion algorithm. Following the aforementioned assumptions, for each novel relation, the algorithm finds its parent with the highest similarity in a top-down paradigm.

Specifically, for each novel relation, starting from the existing root relations, we iteratively search the relation with the highest similarity in candidates layer by layer. In each layer, the search candidates are obtained by the child relations of the search result in the previous layer. The search process terminates if the similarity decreases compared to the previous layer. The extracted relation will be inserted as the child of the most similar relation, or cast as a singleton if the highest similarity is lower than a threshold, where a higher expansion threshold will lead to more singleton relations. The procedure is shown in Algorithm 1, and we refer readers to experiments for a detailed example. In practice, the similarity between a novel relation and an existing relation is given by the average connection between their prototypes as follows:

$$S(r_i, r_j) = \frac{\sum_{v_1 \in P_i} \sum_{v_2 \in P_j} w(\boldsymbol{v}_1, \boldsymbol{v}_2)}{|P_i| \cdot |P_j|} \cdot \sqrt{1 + |P_j^s|}, \quad (7)$$

Algorithm 1 Top-Down Hierarchy Expansion

Require: r: A novel relation
Require: λ_W: Expansion threshold
Init search candidates C = root relations of trees
Init highest similarity in previous layer $W = 0$
while C not empty **do**
  Search relation $\hat{c} = \arg\max_{c \in C} S(r, c)$
  if $S(r, \hat{c}) > W$ **then**
    // Move to the next layer
    Update highest similarity $W = S(r, \hat{c})$
    Update search candidates C = children of $\hat{c}$
  else
    Stop searching
if $W \geq \lambda_W$ **then**
  Expand r as child of $\hat{c}$
else
  Cast r as singleton relation

where r_i is a novel relation and r_j is an existing relation, P_i and P_j are the corresponding relation prototypes, and $|P_j^s|$ refers to the number of all descendant relations of r_j. In experiments, we find that relations containing more descendant relations in hierarchy tend to exhibit lower average connections with novel relations, due to the margins between the contained descendant relations. By introducing $\sqrt{1 + |P_j^s|}$, we balance the connection strength and encourage the model to explore wider and deeper hierarchies.

The reason for expanding hierarchy with a top-down paradigm is threefold: (1) The coarse-to-fine-grained hierarchy expansion procedure is bioplausible, as suggested by cognitive neuroscience studies (Tenenbaum et al., 2011). (2) The decision making procedure following the existing hierarchy structure is interpretable. (3) It can achieve better efficiency since the unlikely branches are pruned in the early search stage.

4 Experiments

To verify the effectiveness of hierarchical information and OHRE, we conduct comprehensive experiments on relation clustering and hierarchy expansion on two real-world datasets. We also conduct a detailed analysis of OHRE to provide a better understanding of our framework. We refer readers to the appendix for more implementation details.

4.1 Dataset

Following previous works (Wu et al., 2019; Hu et al., 2020), we evaluate our framework on FewRel (Han et al., 2018b) and New York Times Freebase (NYT-FB) dataset (Marcheggiani and

Titov, 2016). However, the original random data splits are not suitable to benchmark open hierarchical relation extraction task, since the test sets do not well cover different topologies in relation hierarchy. In the test sets, for a majority of relations, their parent relations are not labeled with sentences in the dataset, making them singleton relations. It is desirable to include more diverse and challenging relations with complex topologies in the test sets. Thus we re-split these two datasets to better approximate and provide benchmarks for real-world needs. Considering applications where only incomplete relation hierarchies are available, we only use partial hierarchy from KBs, removing the hierarchy of relations beyond the train sets.

FewRel Hierarchy. FewRel (Han et al., 2018b) is a supervised dataset created from Wikipedia and Wikidata. Following Wu et al. (2019), the train set includes 64 relations where each relation has 700 instances. The development set and test set share 16 relations, and each set has $1,600$ instances. We exchange relations from the original train and test set to include three relation typologies in test set: (1) single relation without a parent (6 relations), (2) relation with a parent in train set (8 relations), and (3) relation with a parent in test set (2 relations). We call this dataset FewRel Hierarchy.

NYT-FB Hierarchy. NYT-FB (Marcheggiani and Titov, 2016) is a distantly supervised dataset created from New York Times and Freebase. Following Simon et al. (2019), we filter out sentences with non-binary relations. The train set includes 212 relations with $33,992$ instances. The development set and test set share 50 relations, and have $3,835$ and $3,858$ instances respectively. Each relation in development set and test set has at least 10 instances. We call this dataset NYT-FB Hierarchy.

4.2 Experimental Settings

We introduce two task settings and corresponding evaluation metrics. (1) Relation clustering setting is widely adopted in previous OpenRE works to evaluate the ability of clustering novel relations (Marcheggiani and Titov, 2016; Wu et al., 2019). (2) We also design the hierarchy expansion setting to thoroughly test the ability of OpenRE models in expanding existing relation hierarchies.

4.2.1 Relation Clustering Setting

Relation clustering is a traditional OpenRE setting, where models are required to cluster instances into different groups representing new relations.

Baselines. We compare OHRE with state-of-the-art OpenRE baselines. (1) Relational Siamese Network augmented with conditional entropy and virtual adversarial training (**RSN-CV**) (Wu et al., 2019) is the state-of-the-art OpenRE method that transfers relational knowledge from labeled data to discover relations in unlabeled data. (2) **SelfORE** (Hu et al., 2020) utilizes self-training to iteratively learn relation representations and clusters. (3) HAC with re-weighted word embeddings (**RW-HAC**) (Elsahar et al., 2017) is the state-of-the-art rich feature-based method. RW-HAC first extracts rich features, such as entity types, then reduces feature dimension via principal component analysis, and finally clusters the features with HAC. (4) Discrete-state variational autoencoder (**VAE**) (Elsahar et al., 2017) optimizes a relations classifier via reconstruction signals, with rich features including dependency paths and POS tags.

Evaluation Metrics. Following Wu et al. (2019); Hu et al. (2020), we adopt instance-level evaluation metrics to evaluate relation clustering, including B^3 (Bagga and Baldwin, 1998), V-measure (Rosenberg and Hirschberg, 2007) and Adjusted Rand Index (ARI) (Hubert and Arabie, 1985). We refer readers to the appendix for more detailed descriptions about the evaluation metrics.

4.2.2 Hierarchy Expansion Setting

In this setting, models are required to first cluster novel relations, and then further add the extracted relations into the existing hierarchy in train set.

Baselines. To the best of our knowledge, there are no existing OpenRE methods designed to directly expand an existing relation hierarchy. We design two strong baselines based on state-of-the-art OpenRE architectures. (1) RW-HAC for hierarchy expansion (**RW-HAC-HE**) links each novel relation cluster given by RW-HAC to the existing relation cluster with the global highest the Ward's linkage score. The novel relation will be a singleton if the highest score is less than a threshold. (2) RSN-CV for hierarchy expansion (**RSN-CV-HE**) obtains clusters using RSN-CV, and links them to the hierarchy using our top-down expansion algorithm. Here without confusion, we omit the -HE suffixes in model names in the experiment results.

Evaluation Metrics. We adopt two metrics to evaluate on cluster-level (1) how well a predicted cluster matches the golden cluster by matching metric (Larsen and Aone, 1999), and (2) how well

Dataset	Model	B^3			V-measure			ARI
		F1	Prec.	Rec.	F1	Hom.	Comp.	
FewRel Hierarchy	VAE (Marcheggiani and Titov, 2016)	23.0	14.2	61.4	24.1	17.7	37.9	4.9
	RW-HAC (Elsahar et al., 2017)	32.7	28.0	39.4	39.7	36.0	44.4	12.4
	SelfORE (Hu et al., 2020)	60.6	60.1	61.1	70.1	69.5	70.7	54.6
	RSN-CV (Wu et al., 2019)	63.8	57.4	71.7	72.4	68.9	76.2	54.2
	OHRE	**70.5**	64.5	77.7	**76.7**	73.8	79.9	**64.2**
NYT-FB Hierarchy	VAE (Marcheggiani and Titov, 2016)	25.2	17.6	44.4	35.1	28.2	46.3	10.5
	RW-HAC (Elsahar et al., 2017)	35.0	43.3	29.4	58.9	61.7	56.3	28.3
	SelfORE (Hu et al., 2020)	38.1	42.6	34.5	59.0	60.7	57.5	30.4
	RSN-CV (Wu et al., 2019)	38.9	26.3	74.2	44.1	74.3	55.4	26.2
	OHRE	**43.8**	31.4	72.3	**60.0**	49.9	75.3	**31.9**

Table 1: Relation clustering results on two datasets (%).

Model	B^3 F1	V-F1	ARI
RSN-CV	63.8	72.4	54.2
w/o VAT	53.3	65.0	43.2
OHRE	**70.5**	**76.7**	**64.2**
w/o Dynamic Margin	68.9	76.1	63.5
w/o Curriculum Train	68.5	75.7	62.1
w/o Pair-wise VAT	58.3	68.8	49.5

Table 2: Ablation results on FewRel Hierarchy (%).

the predicted cluster links to the golden position in hierarchy by taxonomy metric (Dellschaft and Staab, 2006). We also report two overall evaluation metrics that consider both relation clustering and hierarchy expansion results. Specifically, we report the arithmetic mean and harmonic mean of matching F1 and taxonomy F1.

4.3 Relation Clustering Results

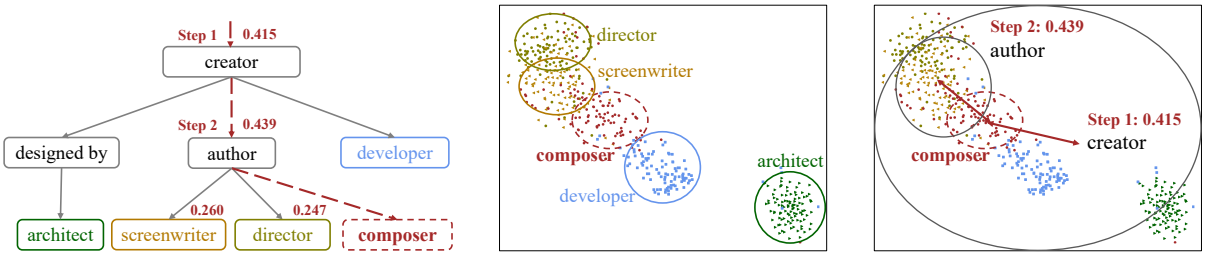
Main Results. Table 1 shows relation clustering results on two datasets, from which we observe that: (1) OHRE outperforms state-of-the-art models by a large margin, e.g., with 6.7%, 4.3%, 9.6% improvements in B^3, V-measure, and ARI respectively on FewRel Hierarchy. Compared with unsupervised methods, the performance gap is even greater, e.g., more than 30% in B^3 on FewRel Hierarchy. This shows that OHRE can effectively leverage existing relation hierarchy for better novel relation clustering. (2) The improvements of OHRE are consistent in both supervised FewRel Hierarchy dataset and distantly supervised NYT-FB Hierarchy dataset. This indicates that the representation learning and relation clustering procedure of OHRE is robust to noisy relation labels and long-tail relations in different domains. We note that although our model adopts CNN as the relation encoder, it outperforms SelfORE equipped with BERT (Devlin et al., 2019). We expect it would be beneficial to enhance the relation representations in OHRE with pre-trained language models, and we leave it for future work.

Ablation Study. We conduct ablations to investigate the contribution of different components, as shown in Table 2. For fair comparisons, we also ablate virtual adversarial training from RSN-CV (Wu et al., 2019). Experimental results show that all components contribute to the final performance. This shows that hierarchical information from existing relations can provide transferable guidance for novel relation clustering. The performance drops most significantly when removing pair-wise virtual adversarial training, indicating the importance of space smoothing to the generalization of OHRE.

4.4 Hierarchy Expansion Results

Main Results. Table 3 shows the results of hierarchy expansion, from which we observe that: (1) OHRE outperforms strong baselines on hierarchy expansion. Compared to baselines, OHRE achieves higher match F1, which indicates that relations extracted by OHRE can be better aligned with golden relations on cluster-level. Moreover, the advantage in taxonomy F1 shows that OHRE can better add the extracted relations in the existing hierarchy. The reasonable overall result shows the potential of OHRE in real-world open hierarchical relation extraction applications. (2) We also conduct hierarchy expansion experiments with golden novel clusters. However, experiment results show no obvious improvements for all models. Particularly, we note that while RW-HAC and RSN-CV achieve seemingly reasonable performance, they always cast novel relation as a singleton and are unable to add the relation to the right place in hierarchy.[4]

[4]The proportion of singleton relations is 37.5%.



(a) The expansion recommendation by OHRE on an existing relation hierarchy. The average connection score at each time step is shown.

(b) OHRE first clusters novel relations from open-domain corpora, and learns relation prototypes.

(c) OHRE then expands relation hierarchy based on relation prototypes in a top-down paradigm.

Figure 3: OHRE workflow in expanding an existing hierarchy with novel relations, and t-SNE visualization on FewRel Hierarchy. Relations with labeled instances in the dataset are marked in color. Relations in existing hierarchy are marked with solid lines, and novel relations are marked with dashed lines. Best viewed in color.

Dataset	Method	Golden Cluster	Match F1	Match Prec.	Match Rec.	Taxonomy F1	Taxonomy Prec.	Taxonomy Rec.	Arith. F1	Harm. F1
FewRel Hierarchy	RW-HAC		33.2	33.9	37.6	37.5	37.5	37.5	35.3	35.2
	RSN-CV		69.6	63.7	85.8	34.5	38.5	31.3	52.0	46.1
	OHRE		**78.5**	73.6	88.4	**53.3**	57.1	50.0	**65.9**	**63.5**
	RW-HAC					43.8	43.8	43.8	71.9	60.9
	RSN-CV	✓		N/A		37.5	37.5	37.5	68.8	54.5
	OHRE					**57.4**	62.5	53.1	**78.7**	**73.0**
NYT-FB Hierarchy	RW-HAC		29.6	34.3	34.0	10.1	8.7	12.0	19.8	15.0
	RSN-CV		45.1	33.2	83.1	10.5	15.2	8.0	27.8	17.0
	OHRE		**51.7**	42.7	76.2	**22.3**	23.9	21.0	**37.0**	**31.2**
	RW-HAC					20.0	16.7	25.0	60.0	33.3
	RSN-CV	✓		N/A		13.0	16.0	11.0	56.5	23.1
	OHRE					**23.0**	23.0	23.0	**61.5**	**37.4**

Table 3: Hierarchy expansion results. Golden cluster indicates the golden relation clusters are given, in which case matching metric for relation clustering is not applicable. Arith: arithmetic mean, Harm: harmonic mean.

	Relation Clustering sgl.	Relation Clustering p-trn.	Relation Clustering p-tst.	Hierarchy Expansion sgl.	Hierarchy Expansion p-trn.	Hierarchy Expansion p-tst.
RW-HAC	31.6	35.0	42.8	**60.0**	0.0	0.0
RSN-CV	67.1	77.8	**64.4**	58.8	0.0	0.0
OHRE	**75.2**	**84.6**	53.9	58.8	**36.4**	0.0

Table 4: Relation clustering (B^3 F1) and hierarchy expansion (Taxonomy F1) results on relations in different hierarchy topologies. sgl.: relations without a parent, p-trn.: parent in train set, p-tst.: parent in test set.

This is because the inconsistent instance representations within each golden cluster will mislead the expansion procedure on cluster-level, which shows integrating hierarchy information into relation representations is of fundamental importance to hierarchy expansion. Besides, the results also show the necessity of re-splitting FewRel to include more hierarchy topologies in test set for better benchmark.

Zoom-in Study. To better understand the performance of models on hierarchy expansion, we divide the relations according to their hierarchy topologies and report the performance on FewRel Hierarchy. Table 4 shows the results on three topologies, including (1) single relations without parents (sgl.), (2) relations with parents in train set (p-trn.), and (3) relations with parents in test set (p-tst.). The results show that although models achieve reasonable performance on clustering in all three topologies, they struggle on hierarchy expansion, especially on relations with parents. In comparison, OHRE can handle some relations with parents in train set. However, there is still ample room for improvement. This shows hierarchy expansion is challenging, and we leave further research for future work.

4.5 Case Study

To intuitively show how OHRE expands an existing hierarchy with novel relations from open-domain corpora, we visualize the workflow of OHRE on relation *composer*, as shown in Figure 3. The average connection score increases as the expansion proce-

dure progress from top to down in hierarchy. The expansion procedure terminates when the connection score decreases. The process is not only better aligned with real-world needs, but also provides better interpretability in decision making.

5 Conclusion

In this work, we make the first attempt to address bidirectional connections between OpenRE and relation hierarchy. In the future, we believe the following directions worth exploring: (1) We use a heuristic method to add new relations into hierarchies based on local similarities between relations. In future, more advanced methods can be designed to model the global interaction between new relations and hierarchy, and learn to effectively add the novel relations. (2) We conduct relation representation learning and hierarchy expansion in a pipeline. In the future, end-to-end models can be developed to jointly optimize these important phases for better open hierarchical relation extraction results.

6 Acknowledgement

This work is supported by the National Key Research and Development Program of China (No. 2020AAA0106501). Yao is also supported by 2020 Tencent Rhino-Bird Elite Training Program.

References

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *Proceedings of ACL-IJCNLP*, pages 344–354. ACL.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer.

Amit Bagga and Breck Baldwin. 1998. Entity-based cross-document coreferencing using the vector space model. In *Proceedings of ACL-COLING*, pages 79–85. ACL.

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *Proceedings of IJCAI*, pages 2670–2676. ACM.

Michele Banko and Oren Etzioni. 2008. The tradeoffs between open and traditional relation extraction. In *Proceedings of ACL: HLT*, pages 28–36. ACL.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of ICML*, page 41–48. ACM.

Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250.

Lei Cui, Furu Wei, and Ming Zhou. 2018. Neural open information extraction. In *Proceedings of ACL*, pages 407–413. ACL.

Klaas Dellschaft and Steffen Staab. 2006. On how to perform a gold standard based evaluation of ontology learning. In *Proceedings of ISWC*, pages 228–241. Springer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL: HLT*, pages 4171–4186. ACL.

Hady Elsahar, Elena Demidova, Simon Gottschalk, Christophe Gravier, and Frederique Laforest. 2017. Unsupervised open relation extraction. In *Proceedings of ESWC*, pages 12–16. Springer.

Xu Han, Pengfei Yu, Zhiyuan Liu, Maosong Sun, and Peng Li. 2018a. Hierarchical relation extraction with coarse-to-fine grained attention. In *Proceedings of EMNLP*, pages 2236–2245. ACL.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018b. FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of EMNLP*, pages 4803–4809. ACL.

Xuming Hu, Lijie Wen, Yusong Xu, Chenwei Zhang, and Philip Yu. 2020. SelfORE: Self-supervised relational feature learning for open relation extraction. In *Proceedings of EMNLP*, pages 3673–3682. ACL.

Zhiting Hu, Poyao Huang, Yuntian Deng, Yingkai Gao, and Eric Xing. 2015. Entity hierarchy embedding. In *Proceedings of ACL-IJCNLP*, pages 1292–1300. ACL.

Lawrence Hubert and Phipps Arabie. 1985. Comparing partitions. *Journal of classification*, 2(1):193–218.

Bjornar Larsen and Chinatsu Aone. 1999. Fast and effective text mining using linear-time document clustering. In *Proceedings of KDD*, page 16–22, New York, NY, USA. ACM.

Claudia Leacock and Martin Chodorow. 1998. Combining local context and wordnet similarity for word sense identification. *WordNet: An electronic lexical database*, 49(2):265–283.

Dekang Lin and Patrick Pantel. 2001. Dirt @sbt@discovery of inference rules from text. In *Proceedings of KDD*. ACM Press.

Diego Marcheggiani and Ivan Titov. 2016. Discrete-state variational autoencoders for joint discovery and factorization of relations. *TACL*, 4:231–244.

Takeru Miyato, Andrew M. Dai, and Ian J. Goodfellow. 2017. Adversarial training methods for semi-supervised text classification. In *Proceedings of ICLR*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of EMNLP*, pages 1532–1543. ACL.

Simone Paolo Ponzetto and Michael Strube. 2007. Knowledge derived from wikipedia for computing semantic relatedness. *JAIR*, 30:181–212.

Andrew Rosenberg and Julia Hirschberg. 2007. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of EMNLP-CoNLL*, pages 410–420. ACL.

Juho Rousu, Craig Saunders, Sandor Szedmak, and John Shawe-Taylor. 2005. Learning hierarchical multi-category text classification models. In *Proceedings of ICML*, pages 744–751. ACM.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of CVPR*, pages 815–823. IEEE Computer Society.

Étienne Simon, Vincent Guigue, and Benjamin Piwowarski. 2019. Unsupervised information extraction: Regularizing discriminative approaches with relation distribution losses. In *Proceedings of ACL*, pages 1378–1387. ACL.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *JMLR*, 15(56):1929–1958.

Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. 2018. Supervised open information extraction. In *Proceedings of NAACL: HLT*, pages 885–895. ACL.

Joshua B. Tenenbaum, Charles Kemp, Thomas L. Griffiths, and Noah D. Goodman. 2011. How to grow a mind: Statistics, structure, and abstraction. *Science*, 331(6022):1279–1285.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57:78–85.

Kilian Q Weinberger and Olivier Chapelle. 2009. Large margin taxonomy embedding for document categorization. In *Advances in NeurIPS*, pages 1737–1744. Curran Associates, Inc.

Ruidong Wu, Yuan Yao, Xu Han, Ruobing Xie, Zhiyuan Liu, Fen Lin, Leyu Lin, and Maosong Sun. 2019. Open relation extraction: Relational knowledge transfer from supervised data to unsupervised data. In *Proceedings of EMNLP-IJCNLP*, pages 219–228. ACL.

Ruobing Xie, Zhiyuan Liu, and Maosong Sun. 2016. Representation learning of knowledge graphs with hierarchical types. In *Proceedings of IJCAI*, pages 2965–2971. IJCAI/AAAI Press.

Limin Yao, Aria Haghighi, Sebastian Riedel, and Andrew McCallum. 2011. Structured relation discovery using generative models. In *Proceedings of EMNLP*, pages 1456–1466. ACL.

Limin Yao, Sebastian Riedel, and Andrew McCallum. 2012. Unsupervised relation discovery with sense disambiguation. In *Proceedings of ACL*, pages 712–720. ACL.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING*, pages 2335–2344. ACL.

Ningyu Zhang, Shumin Deng, Zhanlin Sun, Guanying Wang, Xi Chen, Wei Zhang, and Huajun Chen. 2019. Long-tail relation extraction via knowledge graph embeddings and graph convolution networks. In *Proceedings of NAACL: HLT*, pages 3016–3025. ACL.

A Implementation Details

In this section, we introduce hyperparameters and important bounds in relation representation learning and in relation hierarchy expansion respectively. All hyperparameters are selected by grid search on the development set. Moreover, we report the average training time and the number of parameters.

Representation Learning Hyperparameters. In embedding layer, we use 50-d GloVe (Pennington et al., 2014) word embeddings and 2 randomly initialized 5-d position embeddings, and all the embeddings are trainable. The convolution kernel size is 3, relation embedding size is 64 selected from $\{64, 128, 256, 512\}$, and λ_d in representation learning is 0.7 selected from $\{0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$. We apply dropout (Srivastava et al., 2014) after embedding layer with dropout rate 0.2, and L2 regularization on the convolutional and fully connected layer with hyperparameters 2e-4 and 1e-3 respectively. During training, the batch size is 64 selected from $\{16, 32, 64, 128\}$. For each batch, we randomly sample 4 relation types, each with 16 instances. Besides, hierarchical curriculum learning strategy lasts 100 batches in the first epoch to warm up the model parameters. In pair-wise virtual adversarial training strategy, we first generate perturbation vector δ_1 for each instance representation $\boldsymbol{v}$, where the value in each dimension follows a uniform distribution of range $[0, 1)$. Then the perturbation vector δ_1 is scaled such that its L2 norm is 0.02. We add δ_1 to the instance feature, and compute the worst-case perturbation δ_2 based on the gradient. Finally δ_2 is scaled to 0.02 in L2 norm, and added to the feature of the instance to obtain $\tilde{\boldsymbol{v}}$.

Hierarchy Expansion Hyperparameters. In relation clustering process, Louvain (Blondel et al., 2008) algorithm will not take the similarity between instances less than the threshold 0.5 into account. The instance of novel relation prototypes having less than 5 instances will be moved to their closest neighbors based on the average connection weight. During hierarchy expansion, the thresholds for singleton relations in top-down expansion and RW-HAC-HE are 0.2 and 0.1 respectively.

B Evaluation Metrics

In this section, we provide details of evaluation metrics in two settings.

Relation Clustering Setting. Following previous works (Wu et al., 2019; Hu et al., 2020), we adopt instance-level evaluation metrics, including B^3, V-measure and Adjusted Rand Index.

(1) B^3. For each instance in test set, B^3 computes its precision and recall by comparing the cluster containing the instance in prediction results and the cluster containing the instance in golden answer. After that, B^3 averages the precision and recall of each instance and produces a harmonic mean. (2) V-measure. Similarly, V-measure (Rosenberg and Hirschberg, 2007) is another instance-based measurement that further introduces conditional entropy, which asks for the higher requirement of the purity of clusters. Compare to B^3, the existence of a few wrong instances in a relatively pure cluster decreases more score to punish clustering results. Meanwhile, the V-measure F1 calculates the harmonic mean of homogeneity and completeness. (3) Adjusted Rand Index. ARI (Hubert and Arabie, 1985) counts all pair-wise assignments in the same or different groups to measure the similarity of predicted and golden clusterings. Random node assignment makes ARI be 0, and the maximum of ARI is 1, which means the perfect result. Compared to the previous two metrics, ARI is less sensitive since it won't be influenced by an extreme sub-value like precision or homogeneity.

Hierarchy Expansion Setting. To bridge the predicted clusters with real relations, we first match each predicted cluster to the golden cluster then cast it as a prototype for hierarchy position evaluation. We borrow two metrics to evaluate how well a predicted cluster matches the golden cluster, and how well the predicted cluster links to the golden position in hierarchy on cluster-level.

(1) Matching Metric. Similar to Larsen and Aone (1999), we try to match each predicted cluster to one golden relation with whom the predicted cluster has the highest F1 score on cluster-level. Note that different from the original measurement, the golden relation can be matched once only. For each paired novel cluster and golden relation, we calculate precision, recall, and F1 score, and finally weighted sum up based on the number of instances.

(2) Taxonomy Metric. Taxonomy metric was first proposed to evaluate taxonomy structure (Dellschaft and Staab, 2006). After matching predicted clusters to golden relations, for each predicted cluster, we use taxonomy metric to compare the position of this predicted cluster and the position of the corresponding golden relation in

hierarchy. Assume position p in hierarchy is characterized by the union of all its ancestors and descendants $u(p)$. Denote r_g as the golden position and r_p as the predicted position of relation r in the hierarchy, respectively. The precision is defined as follows,

$$\text{Prec.} = \frac{1}{|P|} \sum_{r \in P} \frac{|u(r_p) \cap u(r_g)|}{|u(r_p)|}, \tag{8}$$

where P are the predicted relation clusters. After symmetrically calculating taxonomy recall, we can get taxonomy F1 by their harmonic average.

(3) Overall Evaluation Metric. To give a global evaluation of open hierarchical relation extraction problem, we propose the Overall Evaluation Metric. It simply combines the matching metric and taxonomy metric by arithmetic mean and harmonic mean, to give an overall score that considers both cluster-level performance and taxonomy-level performance.