Natural Language Engineering (2023), **29**, pp. 1126–1149
doi:10.1017/S1351324923000050

CAMBRIDGE UNIVERSITY PRESS

ARTICLE

An end-to-end neural framework using coarse-to-fine-grained attention for overlapping relational triple extraction

Huizhe Su, Hao Wang*, Xiangfeng Luo and Shaorong Xie*

School of Computer Engineering and Science, Shanghai University, Shanghai, China
*Corresponding authors. Email: wang-hao@shu.edu.cn; srxie@shu.edu.cn

(Received 30 May 2021; revised 6 January 2023; accepted 17 January 2023; first published online 21 February 2023)

Abstract

In recent years, the extraction of overlapping relations has received great attention in the field of natural language processing (NLP). However, most existing approaches treat relational triples in sentences as isolated, without considering the rich semantic correlations implied in the relational hierarchy. Extracting these overlapping relational triples is challenging, given the overlapping types are various and relatively complex. In addition, these approaches do not highlight the semantic information in the sentence from coarse-grained to fine-grained. In this paper, we propose an end-to-end neural framework based on a decomposition model that incorporates multi-granularity relational features for the extraction of overlapping triples. Our approach employs an attention mechanism that combines relational hierarchy information with multiple granularities and pretrained textual representations, where the relational hierarchies are constructed manually or obtained by unsupervised clustering. We found that the different hierarchy construction strategies have little effect on the final extraction results. Experimental results on two public datasets, NYT and WebNLG, show that our mode substantially outperforms the baseline system in extracting overlapping relational triples, especially for long-tailed relations.

Keywords: Relational extraction; Relational hierarchy; Multi-granularity information; Long-tail classification

1. Introduction

Relation extraction (RE) extracts the semantic relations between entities in unstructured text. Traditional pipeline approaches ignore the interaction and correlation between entity detection and relational classification and are prone to error propagation (Li and Ji 2014). Recent works (Yu and Lam 2010; Miwa and Sasaki 2014; Ren *et al*. 2017) also show that jointly integrating the information of entities and relations can solve this problem.

However, due to the inherent complexity of language, there may be multiple entities in a sentence, and some relational triples (subject, relation, and object) in a sentence may share one or more entities among themselves in which the triples are called overlapping relational triples (see Figure 1). For example, "But in the fall of 2004, Asia 's broadest economic shoulders, China and Japan, bumped over a pipeline to ship Siberian oil." has two triples (Asia, /location/location/contains, China) and (Asia, /location/location/contains, Japan), which exhibit *SingleEntityOverlap*. Another case is "In Baghdad, Mr. Gates talked to enlisted service members on the second day of his visit to Iraq." The sentence has two triples (Iraq,/location/location/contains, Baghdad) and (Iraq,/location/country/capital, Baghdad), which exhibit *EntityPairOverlap*.

Table 1. Statistics of datasets. Note that a sentence can belong to both the *EPO* class and *SEO* class

	NYT		WebNLG	
Category	Train	Test	Train	Test
Normal	37013	3266	1596	246
EPO	9782	978	227	26
SEO	14735	1297	3406	457
All	56195	5000	5019	703
Relation	24		171	

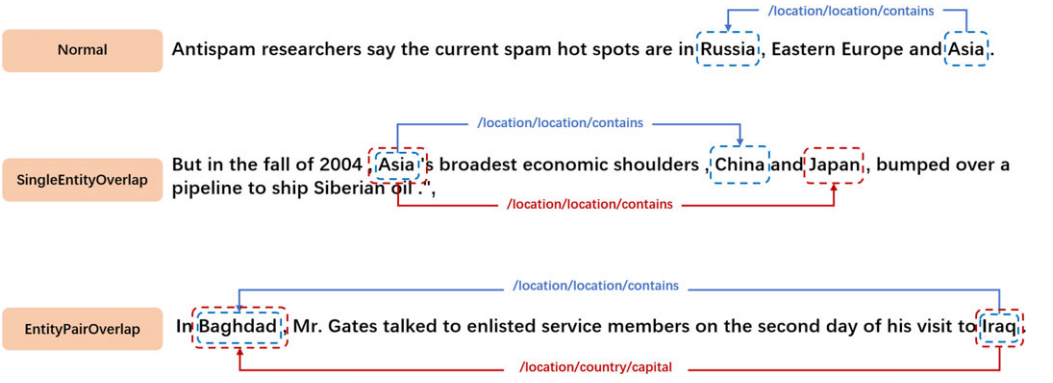

Figure 1. Examples of overlapping relation patterns, e.g., *Normal*, *SingleEntityOverlap (SEO)* and *EntityPairOverlap (EPO)*.

We can find that the overlapping entities in different triples may have different semantic information, for example, in the example of *EntityPairOverlap*, "Baghdad" not only means a region of Iraq but also the capital of "Iraq," making it difficult to extract relationships with overlapping entities.

Sentences in the New York Times (NYT) (Riedel *et al*. 2010) and WebNLG (Gardent *et al*. 2017) datasets commonly contain multiple overlapping relational triples. Statistics of the NYT and WebNLG datasets are described in Table 1. In the NYT training set, the sentences with overlapping triples account for 43% of the sentences, and in the WebNLG training set, they account for 72.3%. It is observed that the sentences with overlapping triples have a proportion that cannot be ignored in the two datasets.

The previously mentioned methods cannot identify overlapping relational triples effectively. Most existing models identify overlapping triples based on two groups of methods, decoder-based and decomposition-based. Decoder-based models rely on the encoder–decoder architecture, where the decoder decodes one word or triple each time (Zeng *et al*. 2018; 2019; Nayak and Ng 2020). Decomposition-based models first distinguish between all the candidate subjects involved in the target relations and then sequentially identify the corresponding object entities and relations for each extracted subject entity (Yu *et al*. 2020; Wei *et al*. 2020).

Despite their success, previous works on overlapping relational extractions still leave much to be desired. Specifically, these methods focus on relational extractions without considering the correlations in the relational hierarchy and ignore the coarse-to-fine-grained semantic information from the category level to the instance level.

We take an example from the NYT (Riedel *et al*. 2010) dataset, as shown in Figure 2, in which the hierarchical structure of relations is manually annotated. Based on this work's (Han

https://doi.org/10.1017/S1351324923000050 Published online by Cambridge University Press

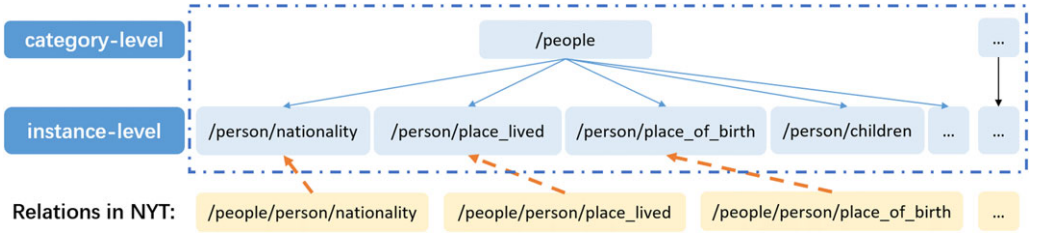

Figure 2. Hierarchical structure of NYT dataset.

et al. 2018) understanding of the relational hierarchy, for example, the relation **/people /person/nationality** in NYT indicates that the relation is labeled under the **people** category. In the relation **/people/person/nationality**, **/person/nationality** contains instance-level information, while **/people** contains the category-level information. There are some other relations under the **people** category, such as **/people/person/place_lived**, **/people/person/place_of_birth** and **/people/person/children**. These relations are closely associated with each other and may have some relevant semantic information.

Therefore, these relational hierarchies may reveal the semantic similarity of relations between entities. McCallum *et al.* (1998) utilized the hierarchies of the classes to improve classification models. Hu *et al.* (2015) and Xie *et al.* (2016) used the hierarchical information of entities from KGs. They demonstrated it to be effective for knowledge graph completion and triple extraction tasks, especially for the data with a long-tailed distribution. Zhang *et al.* (2018) constructed a three-level hierarchical relation structure (HRS) to learn knowledge representation by leveraging rich hierarchical relation information. They achieved significant improvements compared with the baselines on link prediction and triple classification tasks. Han *et al.* (2018) also incorporated the hierarchical information of relations into the relation extraction task, which is especially helpful for extracting those rare relations. Real-world datasets always have a skewed distribution with a long tail with a small number of relation types (i.e., head relation types) occur more frequently, which make up most of the data, and, in contrast, most relations (i.e., long-tail relations) have only a few training instances (Liang *et al.* 2021). Our preliminary experiments also suggest that the hierarchical information from different semantic levels is of great value for relational extraction. For long-tailed rare relation types, coarse-grained semantic knowledge of entities can enhance the model's ability to identify semantically similar relations, whereas fine-grained information can help capture different semantic of the same entity in different overlapping triples.

To handle overlapping relational triples, combined with knowledge of relational hierarchies, in this paper, we propose an end-to-end neural framework using coarse-to-fine-grained attention for overlapping triple extraction. Our model employs the attention mechanism combining the relational hierarchy to incorporate the multi-granularity relational features ranging from the category level to the instance level in the model (Wei *et al.* 2020), which is based on first extracting the subject entities and then extracting the object entities related to the subject and relationship. Our model improves the overlapping relationship extraction by incorporating multi-granular relationship features from the category level to the instance level. Different from existing relational hierarchy methods (Han *et al.* 2018), the hierarchy of relationships is obtained either manually with labels or automatically by clustering. The attention mechanism using the relational hierarchy has two levels. The global attention level captures common features among multiple related relations in a relational cluster, called category-level features. The local attention level focuses on more specific features of the relationships in sentence instances, called instance-level features. Thus, the attention mechanism using relational hierarchy provides both the coarse and the fine granularity of relational features.

https://doi.org/10.1017/S1351324923000050 Published online by Cambridge University Press

We conducted experiments for overlapping relation extraction on two public datasets: NYT (Riedel *et al*. 2010) and WebNLG (Gardent *et al*. 2017). As shown in Table 1, in both datasets, there is a nonnegligible overlapping relationship in the sentences. Nearly, 70% of the relations are long-tailed in the NYT dataset (Zhang *et al*. 2019; Li *et al*. 2020), as is the WebNLG dataset (as shown in Figure 7).

Experimental results show that the proposed method can obtain coarse-to-fine-grained relation information effectively for overlapping relational extraction tasks and outperforms other baseline methods, even compared to the recent state-of-the-art models, especially for long-tail relations.

The research contributions of this paper can be summarized as follows:

(1) We propose an end-to-end neural framework that has an attention mechanism using relational hierarchy to improve the overlapping problem in triple extraction. This attention mechanism enables the capture of semantics from entities to relations by considering the multi-granularity features. Experiments show that our model outperforms previous works and achieves state-of-the-art results on the benchmark datasets.

(2) We obtained the relational hierarchy by using manually annotated relational structures or by using automatic clustering. Our experiments show that the difference between the two methods of constructing the relationship hierarchy is not significant for the final extraction effect. And whether or not category labels are incorporated in the relation names during clustering also has little effect on the final results.

(3) By employing the attention mechanism using relational hierarchy, we fuse multi-granularity relational knowledge to achieve knowledge sharing of similar semantics and knowledge differentiation of different semantics in multiple overlapping triples. Further analysis shows that our model not only improves the extraction of overlapping relations but also improves the triple extraction of long-tail relations.

2. Related work

Extracting relational triples from unstructured texts is an important task in information extraction (IE). Early pipeline methods (Mintz *et al*. 2009; Gormley *et al*. 2015) usually suffer from the error propagation problem. They also overlook the strong association between entity recognition and relation extraction. Feature-based models (Yu and Lam 2010; Li and Ji 2014; Miwa and Sasaki 2014; Ren *et al*. 2017) heavily rely on feature engineering and require intensive manual effort. Neural network-based methods (Gupta *et al*. 2016; Zheng *et al*. 2017) can reduce the manual work and often jointly learn the entities and relations. However, they ignore the problem of overlapping relational triples.

To address the overlapping triples problem, researchers have proposed a variety of neural networks (Zeng *et al*. 2018; 2020; 2019; Nayak and Ng 2020; Yu *et al*. 2020; Wei *et al*. 2020; Bekoulis *et al*. 2018; Fu *et al*. 2019; Wang *et al*. 2020; Zheng *et al*. 2021; Zhang *et al*. 2021; 2022). Existing models can be categorized as decoder-based and decomposition-based models. Decoder-based models use an encoder–decoder architecture, in which the decoder extracts one word or one triple at a time, similar to machine translation models. Zeng *et al*. (2018) analyzed three patterns of overlapping triples existing in the task. They propose an encoder–decoder model with a copy mechanism to handle relation triples with overlapping entities but during decoding, this CopyRE model fails to generate multiword entities. As a supplement, CopyMTL (Zeng *et al*. 2020) proposes a multitask learning framework that completes the entities by adding an auxiliary sequence labeling task to CopyRE. Another variation of CopyRE, called OrderCopyRE (Zeng *et al*. 2019), applies reinforcement learning under the encoder–decoder architecture to deal with multiple triples; Nayak and Ng (2020) employs the WDec decoder to extract each word in a sequence, which allows extracting overlapping relation triples with multitoken entities.

https://doi.org/10.1017/S1351324923000050 Published online by Cambridge University Press

Decomposition-based models extract all possible candidate subject entities first and then extract the corresponding object entities and relations according to each extracted entity. ETL-Span (Yu *et al.* 2020) presents a unified sequence labeling framework based on a novel decomposition strategy that can decode triples hierarchically. However, this method can only recognize SEO relations in the sentence and fail to extract EPO triples. To handle EPO cases, CasRel (Wei *et al.* 2020) is a novel cascade binary tagging framework based on the BERT backbone, which first identifies subject entity candidates in a sentence and then extracts the corresponding object entities given the possible relation for each subject entity.

Other types of models, such as MultiHead (Bekoulis *et al.* 2018), distinguish between all the candidate entities first and then formulate the task as a multihead selection problem. GraphRel (Fu *et al.* 2019) utilizes a graph convolutional network (GCN) to extract overlapping relations by splitting entity mention pairs into several word pairs and considering all the pairs for prediction.

However, these models ignore the rich semantic similarity among the relations, regarding each relation as isolated, especially the hierarchical information of those relations. Hierarchical information is widely applied for model enhancement, especially for classification models (Rousu *et al.* 2005; Weinberger and Chapelle 2009; Zhao *et al.* 2011; Bi and Kwok 2011; Xiao *et al.* 2011; Verma *et al.* 2012). Hu *et al.* (2015) learns entity representation by considering the entire entity hierarchy of Wikipedia. Xie *et al.* (2016) uses a hierarchical-type structure to help learn knowledge graph representations. Zhang *et al.* (2018) learns knowledge representation by constructing a three-level hierarchical relational structure that makes use of rich hierarchical relational information. Han *et al.* (2018) uses the hierarchical information of relationships for relation extraction.

Learning from the above works, to improve the performance of overlapping relation extraction, we design an attention mechanism using relational hierarchy to obtain the multi-granularity semantic features. We have two approaches in obtaining the relationship hierarchy, using manual annotation or automatic clustering. Our model applies the attention mechanism using relational hierarchy to capture the coarse-to-fine-grained semantic features of multiple overlapping triples in a sentence, which can improve the extraction of overlapping triples, as well as triples with long-tail relations.

3. Framework

In this section, we elaborate on the framework of the model, which integrates hierarchical relational information based on decomposition-based models. Then, we describe every part of the model in detail.

Given a sentence $\boldsymbol{s}$, we adopt our models to extract all the triples (subject, relation, and object) whether they are *normal* or *EPO*, *SEO* type. Learn from Casrel (Wei *et al.* 2020), the basic idea is to model relations as functions that map subjects to objects. In other words, we extract all the subjects in the input sentence first and then select all the object entities related to the subject entity and relations. The target relations here are all in the set R. As illustrated in Figure 3, the overall framework of our model includes the BERT encoder, Subject Recognizer, Hierarchical Relation, Information Infusion, and Object Recognizer. The BERT encoder encodes the sentences to obtain a semantic representation. The Subject Recognizer is used to obtain all the subject entities in the sentences. In the Hierarchical Relation Module, there are two levels of relational hierarchy including category level and instance level, which are obtained using manual annotation or automatic clustering, and then the semantic information of the sentence related to each level of the relationship is obtained as multi-granularity semantic information through an attention mechanism. Information Infusion fuses one of the candidate entities, the representation of each token in the sentence, and the multi-granularity semantic information related to the sentence. Finally, the Object Recognizer determines all the corresponding objects for the selected subject entity.

https://doi.org/10.1017/S1351324923000050 Published online by Cambridge University Press

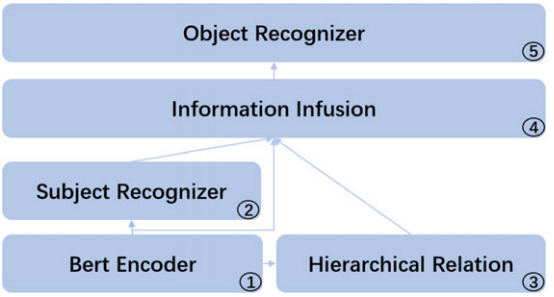

Figure 3. Framework of our model.

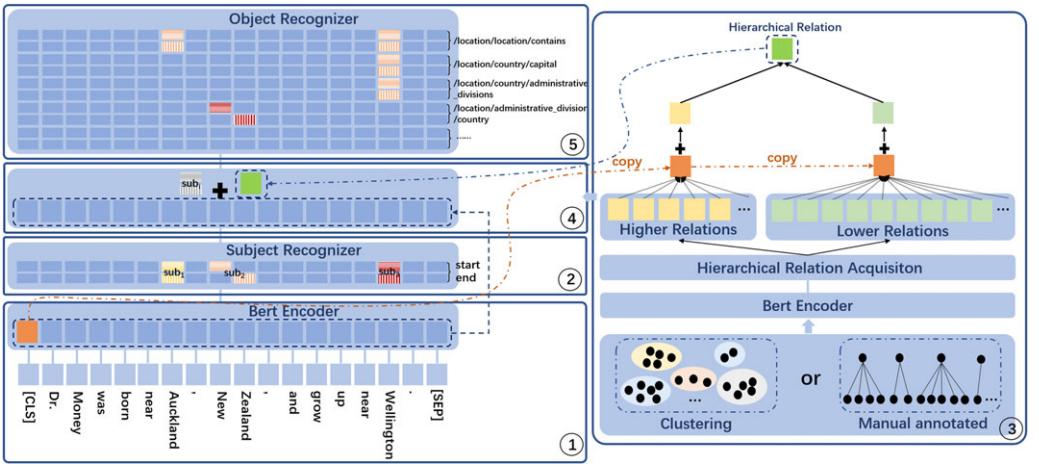

Figure 4. An illustration of our model. The left panel is an overview of our joint extraction model, and the right panel shows the acquisition of the hierarchical structure of relations and how to obtain coarse-to-fine granularity relation information through the attention mechanism. For WebNLG datasets without manually annotated hierarchical relations, the hierarchical structure of the relations is obtained by clustering automatically; for the NYT dataset with a manually annotated relational structure, we use clustering to obtain the hierarchy or utilize the annotated relational structure. Here, we use a two-level structure to unify the number of relational hierarchies of the NYT and WebNLG datasets and take a sentence in the NYT training set as an input example.

3.1. Bert encoder

Given a sentence $\boldsymbol{s} = \{\boldsymbol{w_1}, \boldsymbol{w_2}, \ldots, \boldsymbol{w_n}\}$ as a sequence of tokens, we apply a pretrained BERT model (Devlin *et al.* 2019) to encode the sentence into its corresponding embeddings.

The input representation of each token in BERT is constructed by the summation of the corresponding token, segment, and position embeddings. In our work, the input contains only one sentence, so all its segment IDs are set to zero. In addition, the special tokens [CLS] and [SEP] are placed at the start and end of the sentence. As shown in Figure 4, Module 1 shows the BERT encoder, which extracts feature information from the input sentence $\boldsymbol{s}$. The final hidden vector is $\boldsymbol{S} = \left\{\boldsymbol{T}_{[CLS]}, \boldsymbol{T}_1, \boldsymbol{T}_2, \ldots, \boldsymbol{T}_n, \boldsymbol{T}_{[SEP]}\right\}$, where $\boldsymbol{T}_{[CLS]} \in \mathbb{R}^{\boldsymbol{H}}$ and $\boldsymbol{T}_{[SEP]} \in \mathbb{R}^{\boldsymbol{H}}$ are the final hidden vectors of the special [CLS] and [SEP] tokens, respectively, and $\boldsymbol{T}_i \in \mathbb{R}^{\boldsymbol{H}}$ is the final hidden vector for the i-th token. (H=768)

3.2. Subject Recognizer

The Subject Recognizer aims to recognize all the candidate subjects in the input sentence. The sentence representation encoded by BERT is used as the input of the module. Then, this module

https://doi.org/10.1017/S1351324923000050 Published online by Cambridge University Press

detects whether each token is the start or end position of a subject entity by adopting two binary classifiers and labeling a binary tag as 0 or 1. As shown in Figure 4, Module 2 shows the details of the Subject Recognizer. The start and end positions of the entities in the figure are represented by colored horizontal or vertical line rectangles, respectively; that is, the binary tag of the start or end of the token is 1. Formally, the operations of tagging the start and end positions on each token are Equations (1) and (2), respectively:

$$p_i^{\text{sub\_s}} = \text{Sigmoid}\left(W^s T_i + b^s\right) \tag{1}$$

$$p_i^{\text{sub\_e}} = \text{Sigmoid}\left(W^e T_i + b^e\right) \tag{2}$$

where $W^{(.)}$ is the trainable weight and $b^{(.)}$ is the bias. $p_i^{sub\_s}$ and $p_i^{sub\_e}$ denote the probability of distinguishing the i-th token in the input sentence as the start and end positions of the subject entity, respectively. If the probability exceeds a certain threshold, the corresponding token will be assigned Label 1; otherwise, it will be assigned 0.

The Subject Recognizer optimizes the following likelihood function to identify the span of subjects sub given a sentence representation $\boldsymbol{S}$:

$$p_\theta(sub \mid \boldsymbol{S}) = \prod_{l \in \{\text{sub\_s, sub\_e}\}} \prod_{i=1}^{n} \left(p_i^l\right)^{I\left\{y_i^l=1\right\}} \left(1 - p_i^l\right)^{I\left\{y_i^l=0\right\}} \tag{3}$$

where $I\{x\} = 1$ if x is true and 0 otherwise. $y_i^{sub\_s}$ and $y_i^{sub\_e}$ represent the binary labels of the subject entity start and end positions for the i-th token in $\boldsymbol{s}$, respectively. The parameter n is the length of the input sentence. θ represent the module's parameters, $\theta = \left\{W^s, b^s, W^e, b^e\right\}$.

Since there will be multiple subject entities, we adopt the nearest start–end pair match principle to decide the span of any subject based on the results of the start and end position taggers (Wei *et al.* 2020). For example, as illustrated in Figure 4, "New" is a starting position token, and the ending position tokens are "Zealand" and "Wellington". According to the principle of proximity, the nearest end position token matching the starting position token "New" is "Zealand"; hence, one of the entities in the sentence is "New Zealand".

3.3. Hierarchical relation

We adopt the Hierarchical Relation Module to capture the multi-granularity semantic features of the multiple overlapping relational triples in the input sentence. Accordingly, we need to obtain the relational hierarchy structure of the dataset and the vector representation of all relations of each level.

Because the sentences in both the NYT and WebNLG datasets usually contain multiple relational triples, these two datasets are very suitable as evaluation models for extracting overlapping relational triples. Relations of the NYT dataset are already manually annotated with the hierarchy structure. As shown in Figure 2, for relations such as **/people/person/place_lived**, this relation belongs to the **people** category. In the relation **/people/person/place_lived**, **/person/place_lived** contains instance-level information, while **/people** contains the category-level information. However, relations of the WebNLG dataset do not possess a hierarchical structure annotated manually, for example, **birthPlace** and **capital**. To ensure that the number of levels of the relational structure is consistent with both datasets, and to make full use of the instance-level and category-level hierarchies that NYT has labeled, we unify the relational structure into two levels. Therefore, to obtain the relational hierarchy, we have two approaches, using manual annotation or automatic clustering. Hence, for the NYT dataset, the two-level relationship structure can be constructed not only directly from manual annotation but also through clustering. For the WebNLG dataset, we can have two levels of the relational structure through clustering.

https://doi.org/10.1017/S1351324923000050 Published online by Cambridge University Press

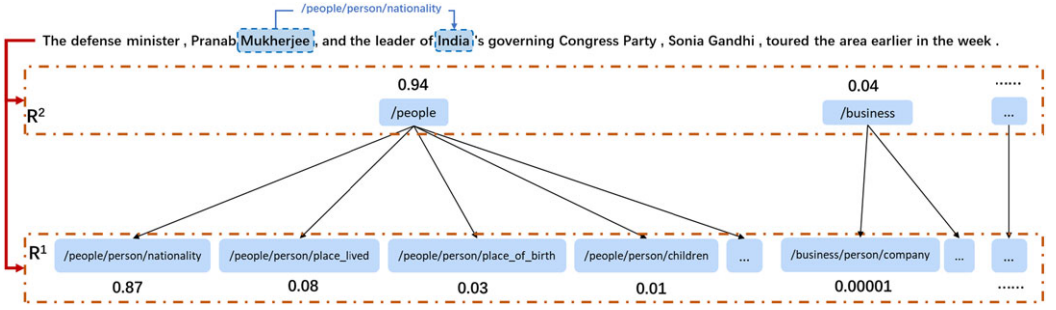

Figure 5. The relationship weight of input sentences at different levels is achieved by the attention mechanism.

In this experiment, we apply the affinity propagation (Frey and Dueck 2007) algorithm to cluster to obtain higher-level relations. Affinity propagation is used to measure the similarity algorithm, which considers all data points as potential exemplars simultaneously. By treating each data point as a node in a network, affinity propagation transmits real-valued messages along the edges of the network until a good set of exemplars and corresponding clusters emerges. Compared with the K-means algorithm, the affinity propagation algorithm does not need to specify the number of final clusters, and the sum of squares error is low. Because of its insensitivity to the initial value, the results of multiple executions are identical. Although the complexity of affinity propagation is higher than that of K-means, in both datasets, the dataset with the most relationships, WebNLG, only has 171 relationships, so the time consumption of affinity propagation is tolerable.

In the following sections, we describe how to obtain the two-level relational hierarchy and the relational representation of each level by clustering, or from the two datasets, and how to use the relational hierarchy to obtain semantic information of the multiple overlapping relational triples through the attention mechanism.

3.3.1. Acquisition of relational hierarchy and the representation of relations

In this section, we focus on how to obtain the two levels' relational hierarchical structure and the relational representation of each level. For datasets that already have a manually labeled relationship hierarchy, we can use the manually labeled relationship hierarchy or obtain it through clustering, such as for NYT; for datasets without a relationship hierarchy, we can only obtain it through clustering, such as for WebNLG. Given a dataset, we define the relational collection R^L as the following equation:

$$R^L = \left\{r_1^L, r_2^L, r_3^L, \cdots, r_{\text{num}_L}^L\right\} \left(L \in N^*\right) \tag{4}$$

where L is the number of levels of the relationship; when L is 1, it denotes the base level. num_L is the number of relationships at the L-th level and r_i^L is the i-th relation of the L-th level. For example, for the NYT dataset with a manually annotated hierarchy, we select two levels from the annotated hierarchy. As shown in Figure 5, taking the relation **/people/person/place_live** as an example, we choose the category of **people** as one of the relations in the higher level, that is, L is 2, then the complete presentation **/people/person/place_live** is one of the relations in the base level, that is, L is 1.

W_i^L denotes the sequence of words in r_i^L, where the length of the sequence is k. For example, the i-th relation r_i^L at the base level R^L in NYT is **/people/person/place_live**, where L is 1. After converting nonalphabetic symbols such as "/" and "_" to spaces, the relation r_i^L consists of a series of words W_i^L **/people/person/place_live**:

$$W_i^L = \left\{w_{i1}^L, w_{i2}^L, w_{i3}^L, \cdots, w_{ik}^L\right\} \tag{5}$$

https://doi.org/10.1017/S1351324923000050 Published online by Cambridge University Press

where w_{iv}^L denotes the v-th word of the i-th relation at the L-th level. For each relation in the base-level, that is, L=1, we apply the feature-based approach without fine-tuning any of the parameters of BERT (Devlin *et al.* 2019) to obtain a representation of each relation T_i^L in the base level:

$$T_i^L = BERT\left(\left\{w_{i1}^L, w_{i2}^L, w_{i3}^L, \cdots, w_{ik}^L\right\}\right) \tag{6}$$

$$T_i^L = \left\{t_{i1}^L, t_{i2}^L, t_{i3}^L, \cdots, t_{ik}^L\right\} \tag{7}$$

where $t_{ij}^L \in \mathbb{R}^{d_w}$ is the final hidden vector corresponding to w_{ij}^L. Then, we obtain the embedding of each relation at the base level by the mapping function, as shown in Equation (8):

$$e_i^L = f\left(T_i^L\right) \tag{8}$$

Here, we apply the "average" or "sum" function as the mapping function. $e_i^L \in \mathbb{R}^{d_w}$ denotes the final embeddings of the i-th relation r_i^L at the L-th level:

$$E^L = \left\{e_1^L, e_2^L, e_3^L, \cdots, e_{\mathrm{num}_L}^L\right\} \tag{9}$$

As shown in Eq. 9, E^L represents the collection of the final embeddings of all relations at the L-th level. For the relationship hierarchy using manual annotation, the process of obtaining the base-level relationship representation and the higher-level relationship representation is the same, as shown in Equations 6–9.

To obtain the relational hierarchy using clustering, we cluster the embeddings of the L-th-level relations E^L and use the affinity propagation clustering algorithm to obtain the hierarchy of the L+1-th-level relations. Thus, we divide E^L into num_{L+1} disjoint clusters $\left\{C_j^L \mid j = \lambda_1^L; \lambda_2^L; \cdots; \lambda_{\mathrm{num}_{L+1}}^L\right\}$, where $C_{j'}^L \cap_{j' \neq j} C_j^L = \emptyset$ and $E^L = \bigcup_{j=1}^{\mathrm{num}_{L+1}} C_j^L$. Correspondingly, we use $\lambda^L = \left(\lambda_1^L; \lambda_2^L; \cdots; \lambda_{\mathrm{num}_{L+1}}^L\right)$ to show the clustering result and denote the cluster label of relation r_j^L by $\lambda_{\mathrm{m}}^L \in \{1, 2, \cdots, num_{L+1}\}$, that is, $e_j^L \in C_{\lambda_{\mathrm{m}}^L}^L$. Therefore, we cluster the base-level relations R^L (L=1) to obtain the relation of the L+1 level with the following equation:

$$\lambda^L = AP\left(E^L\right) \tag{10}$$

Before clustering, we need to perform data normalization and dimensionality reduction. Here, we choose zero-mean normalization to normalize the obtained base-level relation vectors E^L (L=1) and use the PCA dimension reduction method (Tipping and Bishop 1999) to reduce the dimensionality of these high-dimensional relationship features before clustering.

Different choices of mapping functions for the relations at the base level can lead to different embeddings of the obtained relations and therefore affect the clustering results. The clustering results are shown in Figure 6. For the WebNLG dataset, as shown in Figure 6(a) and (b), when the "average" mapping function is used for the base-level relation, the number of clustering results of 171 base-level relations is 22, and when the "sum" mapping function is used, the number of clustering results is 29. Similarly, the NYT dataset, which has a manually annotated relational hierarchy, has different clustering results depending on the chosen mapping function, although the number of clustering results is 10. The results are presented in Figure 6(c) and (d). For example, when the "average" mapping function is used in the WebNLG dataset, the number of clustering results of 171 relationships is 22, of which the relationships in cluster 0 are **1st_runway_SurfaceType**,**LCCN_number**, **birthPlace**, **currentTenants**, **ethnicGroups**, and **runwayLength**.

After obtaining the clustering results of the relationships at the L-th level, it is further necessary to obtain the representation of the relationships at the L+1-th level.

The centroid of each category obtained by the affinity propagation algorithm is the existing data points in the sample, rather than the clustering center obtained by averaging multiple data

https://doi.org/10.1017/S1351324923000050 Published online by Cambridge University Press

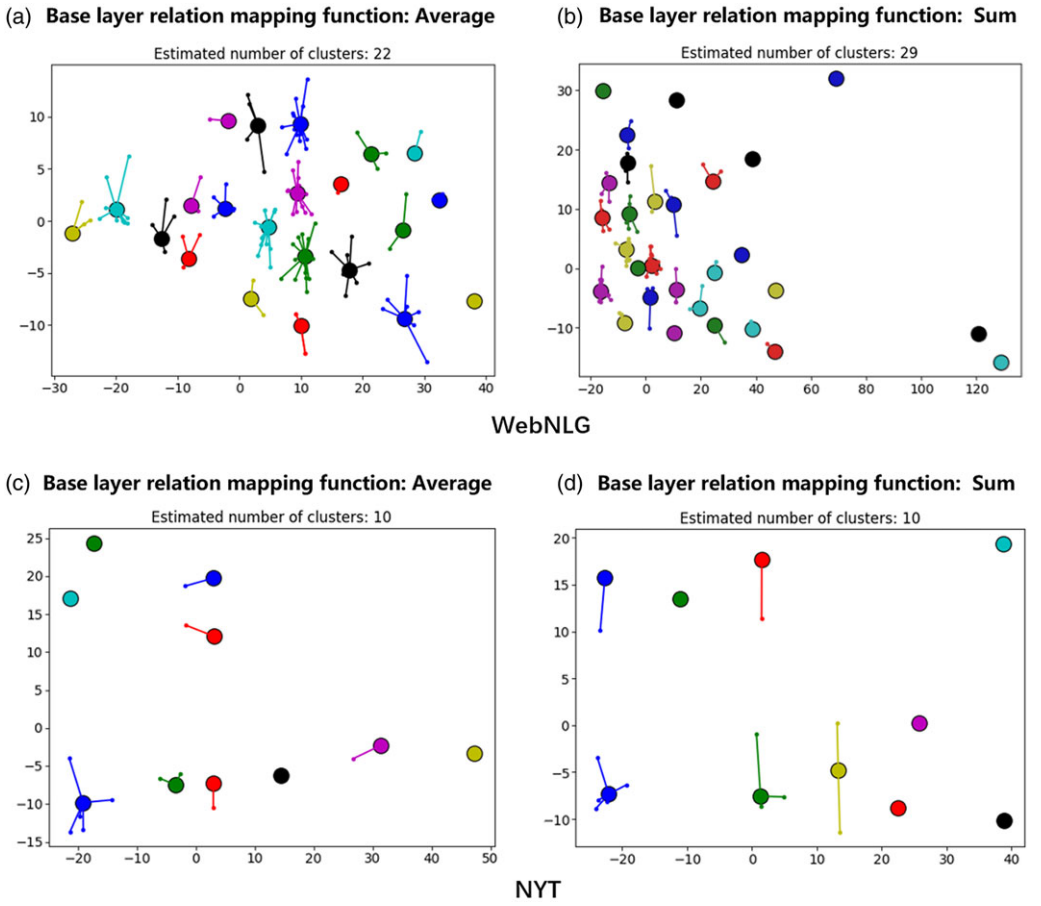

Figure 6. Clustering results of different datasets. (a) and (b) are the clustering results of the WebNLG dataset when the mapping functions of obtaining the overall representation of base-level relations are the "average" and "sum"; when the "average" and "sum" are the mapping functions of the NYT dataset to obtain the overall representation of base-level relations, (c) and (d) are the clustering results.

points. To obtain more of the other characteristics in the category, we do not use the centroid of each category as the representation of higher-level relations. Thus, the representations of all relations for each cluster at the L-th level are passed through the mapping function to obtain the overall representation of each cluster in the L+1-th level, as shown in Eq. 11:

$$e_j^{L+1} = f\left(C_j^L\right) \tag{11}$$

Here, we apply the "average," "max," or "sum" function as the mapping function. $e_j^{L+1} \in \mathbb{R}^{d_w}$ denotes the final embeddings of the j-th relation r_j^{L+1} at the L+1-th level. Then, we can obtain the embeddings of all relations E^{L+1} at the L+1-th level.

3.3.2. Acquisition of multi-granularity relational information through an attention mechanism using the relational hierarchy

We use the final hidden state corresponding to [CLS] as the aggregate representation for the input sentence (Devlin *et al.*, 2019).

https://doi.org/10.1017/S1351324923000050 Published online by Cambridge University Press

Given the representation of the input sentence $\boldsymbol{E_s} = \boldsymbol{T}_{[CLS]}$, the base-level relation E^L and the higher-level relation E^{L+1}, where L is 1, we use an attention mechanism for each level relation to capture the different potential levels of the relational semantics in the input sentence. In the L-th-level relations, we compute the attention score α_i^L for each embedding e_i^L of the relation r_i^L to indicate how much the input sentence semantics are related to this relation. We assign a query vector q_r to the embedding of each relation e_i^L. The formulas for obtaining the relational information associated with the sentence semantics at each level are shown below:

$$h_i^L = q_r^\top W_s e_i^L \tag{12}$$

$$\alpha_i^L = \frac{\exp\left(h_i^L\right)}{\sum_{p=1}^{num_L} \exp\left(h_p^L\right)} \tag{13}$$

$$q_r = E_s \tag{14}$$

$$I^L = \sum_{i=1}^{num_L} \alpha_i^L e_i^L \tag{15}$$

where W_s is the weight matrix. The attention scores α_i^L can be used in Eq. 15 to compute the L-th-level relation information $I^L \in \mathbb{R}^{d_w}$implied in the sentence. For simplicity, we denote such an attention operation as the following equation:

$$I^L = ATT\left(q_r, E^L\right) \tag{16}$$

There are two levels of relations. We can compute the relational information associated with the sentence semantics of the base level I^1 and the higher level I^2 through Eq. 16, respectively.

As shown in Figure 5, the attention score is calculated by the input sentence and the relations in each level. The attention scores are multiplied by the sentence, and the latent relational knowledge of each level contained in the sentence can be obtained by "sum." During the training process, because the base-level relations always suffer from data sparsity, those higher-level relations have more sentences for training than those base-level relations. Thus, the latent relational information in the sentence obtained from higher-level relations contains more knowledge than that obtained from the base-level relations, namely, the knowledge obtained from the global attention can make up for the information captured from the local attention, especially for those long-tail relations.

After obtaining the latent relational knowledge of each level contained in the sentence, we can add them up as the final multi-granularity relational representation:

$$I_{\text{hier}} = Add\left(\left\{I^1, \cdots, I^n\right\}\right) \tag{17}$$

where n is 2 in our experiments. Representation $I_{hier} \in \mathbb{R}^{d_w}$ will be finally infused with other information for recognizing objects. Note that those higher representations I^{L+1} are coarse-grained, and those base representations I^L are fine-grained. Hierarchical attention can accumulate more information than single-level attention, especially for long-tail relations.

3.4. Information fusion

Information fusion is used to fuse the relevant knowledge and provide rich semantic features to the next module, Object Recognizer. We need to extract the corresponding objects in the next module based on the given subject and the knowledge of the multi-granularity relations implied in the sentence. The specific fusion operations for each token are as follows:

$$I_{\text{feature}} = Add\left(I_{\text{hier}}, sub_k\right) \tag{18}$$

$$I_i = Add(I_{\text{feature}}, \boldsymbol{T_i}) \tag{19}$$

https://doi.org/10.1017/S1351324923000050 Published online by Cambridge University Press

where sub_k is the encoded representation vector of the k-th subject detected in the Subject Recognizer module. Note that the subjects are usually composed of multiple tokens, and we need to keep the vector dimensions of I_{hier} and sub_k consistent. Hence, we take the averaged vector representation between the start and end tokens of the k-th subject as sub_k. $I_{feature}$ represents the fusion information of the given k-th subject and the multi-granularity relational knowledge, and I_i denotes the embeddings of the i-th token after fusion with the relevant knowledge.

In this way, the presentation of each token can integrate the coarse-to-fine-grained relational features and provide more semantic knowledge for the next step of object recognition.

3.5. Object Recognizer

The structure of the Object Recognizer module is similar to the Subject Recognizer, and the goal is to recognize all the objects and relations with the subject according to the given subject. As illustrated in Figure 4, it consists of a set of relation-specific Object Recognizer that can obtain the corresponding objects for each detected subject at the same time. The detailed operations of the relation-specific Object Recognizer on each token are as follows:

$$p_i^{\text{obj\_s}} = \text{Sigmoid}\left(W_r^s I_i + b_r^s\right) \tag{20}$$

$$p_i^{\text{obj\_e}} = \text{Sigmoid}\left(W_r^e I_i + b_r^e\right) \tag{21}$$

where $W^{(.)}$ and $b^{(.)}$ are the trainable weight and the bias of the specific relation r that maps the subject to the object. $p_i^{obj\_s}$ and $p_i^{obj\_e}$ represent the probability of identifying the i-th token in the input sentence as the start and end positions of the object, respectively. For each subject, we employ the same decoding process. The Object Recognizer for relation R^L(L=1) optimizes the following likelihood function to recognize the span of objects *obj* given a sentence representation $\boldsymbol{S}$ and a subject ***sub***:

$$p_\theta(obj \mid \boldsymbol{S}, \boldsymbol{sub}, \boldsymbol{r}) = \prod_{l \in \{\text{obj\_s,obj\_e}\}} \prod_{i=1}^{n} \left(p_i^l\right)^{I\left\{y_i^l=1\right\}} \left(1 - p_i^l\right)^{I\left\{y_i^l=0\right\}} \tag{22}$$

where $y_i^{obj\_s}$ and $y_i^{obj\_e}$ are the binary labels of the object start and end positions for the i-th token in $\boldsymbol{s}$, respectively. θ represents the module's parameters, $\theta = \left\{W_r^s, b_r^s, W_r^e, b_r^e\right\}$. For a given subject and corresponding relationship, no object is identified, which means that there is no such relationship between that subject entity and other entities. So, for a "null" object $obj_\emptyset$, the label $y_i^{obj\_s_\emptyset} = y_i^{obj\_e_\emptyset} = 0$ for all i.

Note that the relation is also decided by the output of the Object Recognizer; thus, this module is capable of simultaneously identifying the relations and objects about the subjects detected in the Subject Recognizer. As shown in Figure 4, the relation **/location/administrative_division/country** does not hold between the detected subject "New Zealand" and the candidate object "Wellington," but the relation holds between the detected subject "Wellington" and the candidate object "New Zealand". Consequently, the Object Recognizer for the relation **/location/administrative_division/country** will not extract the span of "Wellington" with the given subject "New Zealand" and outputs the span of the candidate object "New Zealand" when the given subject is "Wellington" instead.

3.6. Training objective

We define the objective as below:

$$\begin{aligned} J_\theta = \sum_{j=1}^{|D|} \Big[& \sum_{\text{sub} \in \text{Tuple}_j} \log p_\theta(sub \mid \boldsymbol{S}) + \sum_{r \in \text{Tuple}_j \mid sub} \log p_{r\theta}(\text{obj} \mid \boldsymbol{S}, \boldsymbol{sub}) \\ & + \sum_{r \in R \setminus \text{Tuple}_j \mid sub} \log p_{r\theta}\left(\text{obj}_\emptyset \mid \boldsymbol{S}, \boldsymbol{sub}\right) \Big] \end{aligned} \tag{23}$$

https://doi.org/10.1017/S1351324923000050 Published online by Cambridge University Press

where D is the training set, one of sentences from D is $\boldsymbol{s}$, a set of potentially overlapping triples $Tuple_j = \{(sub, r, obj)\}$ are in $\boldsymbol{s}$, and $Tuple_j \mid sub$ is the set of triples led by the subject *sub* in $Tuple_j$. $R\backslash$ $\text{Tuple}_j \mid sub$ means all relations except those led by *sub* in $Tuple_j$. Note that all other relations necessarily have no object in the sentence.

4. Experiments

4.1. Datasets

To compare our model with previous work, we use the NYT (Riedel *et al.* 2010) and WebNLG (Gardent *et al.* 2017) datasets for evaluation. The NYT dataset was originally produced by the distant supervision method. It has 24 predefined relation types. The WebNLG dataset is used for natural language generation tasks. The original WebNLG dataset contains 246 predefined relational types, but the relation number of this dataset is 171 in many previous works (Zeng *et al.* 2018; Fu *et al.* 2019; Zeng *et al.* 2019; Yu *et al.* 2020; Nayak and Ng 2020). In our experiments, we use the WebNLG with 171 relations.

4.2. Evaluation

There are two different evaluation metrics selectively adopted from among previous works. The widely used one is the Partial Match, in which an extracted relational triple (subject, relation, and object) is regarded as correct only if the relationship and the start of both the subject and object are all correct. The strict one is an Exact Match where an extracted relational triple (subject, relation, and object) is regarded as correct only if the relationship and the whole span of subject and object are all correct. For example, a gold triple is (Amy Grant, /people/person/place_lived, and Nashville), for Partial Match, the pred triple (Amy, /people/person/place_lived, and Nashville) can be regarded as correct, and the pred triple (Amy Grant, /people/person/place_lived, Nashville) is regarded as correct.

In our implementation, we apply Partial Match as the evaluation metrics. To compare with other models using exact entity matching, we also use exact entity matching to obtain the final results of the best training model in the two datasets. Following the popular choice, we also measure and report the standard micro precision (Prec.), Recall (Rec.), and F1 scores are consistent with all the baselines.

4.3. Implementation details

Before clustering, we use the PCA dimensionality reduction method and n_components is set to 2. We use the base-cased-english BERT model. The number of stacked bidirectional transformer blocks N is 12. The size of the hidden state H is 768. We set the batch size to 6 and the learning rate to 1e-5. We apply the early stop mechanism to prevent model overfitting, which terminates the training course before exceeding 100 epochs. When the performance of the verification set has not been improved for at least seven consecutive periods, we stop the training process. According to previous works, we set the maximum length of the input sentence to 100 and the threshold of the start and end position taggers to 0.5. All the hyperparameters are tuned on the validation set.

4.4. Comparison models

We compare our model with several strong state-of-the-art models.

https://doi.org/10.1017/S1351324923000050 Published online by Cambridge University Press

(1) CopyRE (Zeng *et al*. 2018) adopts a sequence-to-sequence learning model with a copy mechanism, which can extract relevant relational facts from the sentences of these classes.

(2) GraphRel (Fu *et al*. 2019) employs GCNs to better extract hidden features for jointly learning entities and relations.

(3) OrderCopyRE (Zeng *et al*. 2019) adds reinforcement learning to an encoder–decoder model to generate multiple triples, which is an extension of CopyRE.

(4) MrMep (Chen *et al*. 2019) proposes a novel encoder–decoder architecture that includes a binary CNN classifier for identifying all possible relations maintained in the text and multihead attention to extract the entities corresponding to each relation.

(5) HRL (Takanobu *et al*. 2019) applies a hierarchical reinforcement learning (HRL) framework for joint entity and relation extraction.

(6) ETL-Span (Yu *et al*. 2020) decomposes the joint extraction task into HE extraction and TER extraction. The former subtask distinguishes between all subject entities, and the latter identifies the corresponding object entities and relations for each extracted subject entity.

(7) WDec (Nayak and Ng 2020) proposes a representation scheme for relational triples that enables the decoder to generate one word and a pointer network-based decoding approach where an entire triple is generated at every time step.

(8) CasRel (Wei *et al*. 2020) is based on the BERT backbone, which can first extract all the possible subject entities and then identify all the relevant relations and the corresponding object entities.

(9) RSAN (Yuan *et al*. 2020) uses a relation-specific attention network (RSAN) with sequence labeling to jointly extract the entities and relations.

(10) RIN (Kai *et al*. 2020) designs a recurrent interaction network to explicitly capture the intrinsic connections between the entity recognition task and relational classification task.

(11) CGT (Ye *et al*. 2021) introduces a transformer-based generative model for contrastive triple extraction.

4.5. Experimental results and analysis

4.5.1. Main results

Table 2 reports the different results of the relational triple extraction between our model and other baseline models on two datasets, including experimental results using Partial entity matching and Exact entity matching. We present the results of the BERT-based model and non-BERT model separately. We discover that our model outperforms all the baselines in terms of recall and F1. Since relations in the WebNLG dataset have no hierarchical structure, the hierarchical structure of the relations can only be gained through clustering. Consequently, there is just one result on the WebNLG datasets. Take the results of Partial entity matching as an example, our model improves the Rec-score by 2.6% and the F1-score by 0.4% over the best state-of-the-art model (Wei *et al*. 2020) on WebNLG.

Although relations in the NYT dataset have an identical hierarchical structure, we still cluster the base-level relations to obtain the hierarchical structure to make a fair comparison. Hence, there are two results on the NYT dataset. $Our_{Cluster}$ which fuses the hierarchical relation structure obtained by clustering, improves the F1-score by 0.2%, and Our_{Manual} fuses the manually annotated hierarchical relation structure of the dataset, which improves the F1-score by 0.3%. We note that the effect of Our_{Manual} is better than that of $Our_{Cluster}$. This may be caused by the error propagation of the clustering, which causes the relational hierarchical information integrated into the model to also have errors.

https://doi.org/10.1017/S1351324923000050 Published online by Cambridge University Press

Table 2. The main results for partial entity matching and exact entity matching. The datasets labeled ♣ are the results of exact entity matching and vice versa for partial entity matching. The $CasRel_{BERT}$ marked with * is the result of our reimplementation, and the results marked with † are on the validation dataset. The highest scores are marked in bold. These comparison model results are quoted directly from the original papers

	NYT			WebNLG			NYT♣			WebNLG♣		
Model	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
CopyRE	61.0	56.6	58.7	37.7	36.4	37.1	–	–	–	–	–	–
GraphRel	63.9	60.0	61.9	44.7	41.1	42.9	–	–	–	–	–	–
OrderCopyRE	77.9	67.2	72.1	63.3	59.9	61.6	–	–	–	–	–	–
MrMep	–	–	–	–	–	–	77.9	76.6	77.1	69.4	77.0	73.0
HRL	–	–	–	–	–	–	78.1	77.1	77.6	69.5	62.9	66.0
ETL − Span	84.9	72.3	78.1	74.0	91.5	87.6	85.5	71.7	78.0	84.3	82.0	83.1
WDec	**94.5**	76.2	84.4	–	–	–	88.1	76.1	81.7	–	–	–
$CasRel_{LSTM}$	84.2	83.0	83.6	86.9	80.6	83.7	–	–	–	–	–	–
RSAN	–	–	–	–	–	–	85.7	83.6	84.6	80.5	83.8	82.1
RIN	–	–	–	–	–	–	83.9	85.5	84.7	77.3	76.8	77.0
CGT_{BERT}	–	–	–	–	–	–	**94.7**	84.2	89.1	**92.9**	75.6	83.4
$CasRel_{BERT}$	89.7	89.5	89.6	**93.4**	90.1	91.8	–	–	–	–	–	–
$CasRel_{BERT}$*	87.6	90.2	88.9	90.2	90.5	90.4	89.3	88.5	88.9	87.6	85.1	86.3
$Our_{Cluster}$	90.1	89.4	89.8	91.6	**92.7**	**92.2**	89.2	**90.1**	**89.7**	87.9	**85.7**	**86.8**
Our_{Manual}	89.4	**90.4**	**89.9**	–	–	–	89.6	89.6	89.6	–	–	–
$CasRel_{BERT}$*†	88.5	91.3	89.8	91.2	91.3	91.3	89.8	88.8	89.3	88.6	84.5	86.5
$Our_{Cluster}$†	90.9	90.3	90.6	92.2	**93.0**	**92.6**	89.4	**90.3**	89.8	87.6	**88.3**	**87.9**
Our_{Manual}†	90.5	**91.3**	**90.9**	–	–	–	90.0	89.8	**89.9**	–	–	–

4.5.2. Ablation study

To demonstrate the role of the attention mechanism for each level, we remove the attention mechanism one level at a time to see its impact on performance. From these ablations shown in Table 3, we report the results without fusing the relational hierarchy knowledge, which is $CasRel_{BERT}$, fusing only the local-level relational knowledge and fusing only the global-level relational knowledge. We observe that when the relationship information obtained from the local attention is removed, then the F1 score decreases by 0.3% and 0.48% for the NYT dataset for the relational hierarchy obtained from both automatic clustering and manual annotation, respectively, and by 0.56% for the WebNLG dataset. When the relationship information obtained from global attention is removed, the F1 score decreases by 0.45% and 0.88% for the NYT dataset for both the relationship hierarchy obtained by automatic clustering and the relationship hierarchy using manual annotation, respectively, and by 0.76% for the WebNLG dataset. We find that the relational information obtained from the local attention and global attention can provide effective information for the extraction of relational triples, and both are indispensable. Moreover, for multiple overlapping triples with the same entities in a sentence, similar semantic information obtained from global attention is more useful for the extraction of relational triples than different semantic information obtained from local attention.

https://doi.org/10.1017/S1351324923000050 Published online by Cambridge University Press

Table 3. An ablation study of the Partial entity matching results of our model on the validation set. The model that does not combine with knowledge of relational hierarchies is $CasRel_{BERT}$, where our reimplementation is marked by *. The results obtained by $Our_{Cluster}$ or Our_{Manual} in different datasets with the most worked mapping function, that is, $Our_{Cluster}$ uses the "AvgAvg" mapping function in NYT and the "SumAvg" mapping function in WebNLG, while Our_{Manual} uses the "SumAvg" mapping function in NYT

	NYT			WebNLG		
Model	Prec.	Rec.	F1	Prec.	Rec.	F1
$CasRel_{BERT}$*	88.51	91.31	89.89	91.29	91.37	91.33
$Our_{Cluster}$	90.91	90.38	90.65	92.25	93.08	92.67
–local	90.44	90.26	90.35	91.88	92.54	92.11
–global	90.52	89.88	90.20	91.91	91.91	91.91
Our_{Manual}	90.54	91.31	90.92	–	–	–
–local	89.88	91.01	90.44	–	–	–
–global	90.08	90.01	90.04	–	–	–

Table 4. F1-score of sentences with the different overlapping patterns using Partial entity matching. Baselines are all quoted directly from Wei *et al.* (2020) except for the ETL-Span. The results of ETL-Span are reproduced from official implementation

	NYT			WebNLG		
Model	Normal	EPO	SEO	Normal	EPO	SEO
CopyRE	66.0	55.0	48.6	59.2	36.6	33.0
GraphRel	69.6	58.2	51.2	65.8	40.6	38.3
OrderCopyRE	71.2	72.8	69.4	65.4	67.4	60.1
ETL − Span	**88.5**	60.3	87.6	87.3	80.5	91.5
$CasRel_{BERT}$	87.3	92.0	**91.4**	89.4	94.7	92.2
Ours	87.9	**92.3**	91.3	**89.4**	**95.3**	**92.6**

4.5.3. Detailed results on different types of overlapping triples

To verify the ability of our model to extract triples of overlapping relations, we experiment on different types of sentences and compare them with previous works. Table 4 shows the results. For the three different overlapping modes, most models in *normal*, *EPO*, and *SEO* modes show a downward trend. This shows that the extraction of the *SEO* mode is the most difficult, while the extraction of the normal mode is the least difficult. The F1-score of our model not only does not decrease with increasing extraction difficulty but also has a better effect than the $CasRel_{BERT}$ model on the WebNLG dataset.

For the WebNLG dataset, although the F1-score of our model is the same as the state-of-the-art model on the *normal* mode, it improves by 0.6% on the *EPO* mode and 0.4% on the *SEO* mode in terms of the F1-score, over the $CasRel_{BERT}$. For the NYT dataset, our model is only 0.6% and 0.3% higher on the F1-score than the $CasRel_{BERT}$ model on *normal* and *EPO* modes, respectively. This may be because there are 171 relation types in the WebNLG dataset, more than the NYT

https://doi.org/10.1017/S1351324923000050 Published online by Cambridge University Press

Table 5. F1-score of sentences with different triple numbers using Partial entity matching. The data source is the same as Table 4

	NYT					WebNLG				
Model	N=1	N=2	N=3	N=4	$N \geq 5$	N=1	N=2	N=3	N=4	$N \geq 5$
CopyRE	67.1	58.6	52.0	53.6	30.0	59.2	42.5	31.7	24.2	30.0
GraphRel	71.0	61.5	57.4	55.1	41.1	66.0	48.3	37.0	32.1	32.1
OrderCopyR	71.7	72.6	72.5	77.9	45.9	63.4	62.2	64.4	57.2	55.7
ETL − Span	85.5	82.1	74.7	75.6	76.9	82.1	86.5	91.4	89.5	91.1
CasRel$_{BERT}$	88.2	90.3	91.9	94.2	**83.7**	89.3	90.8	94.2	**92.4**	90.9
Ours	**88.7**	**90.6**	**92.1**	**94.4**	83.2	**89.3**	**91.1**	**95.5**	92.1	**91.1**

dataset has with its 24 relation types. Accordingly, when integrating the multi-granularity relation information, the more types of relations in the dataset, the more common knowledge between similar semantics can be captured by the model.

We also validate our model's capability in extracting relational triples from sentences with a different number of triples. The detailed results are presented in Table 5. It can be seen that the performance of most models will decline with an increase in the number of triples in the sentence, while our model has less impact. For the NYT dataset, the F1 scores of our model are higher than those of the state-of-the-art model in N<5 modes except in $N \geq 5$ mode. Similarly, on the WebNLG dataset, our model also has higher F1 scores than the state-of-the-art model in N<4 and $N \geq 5$ modes, except in N=4 mode. This may be because the more relational triples overlap in a sentence, the more demanding it is for the model to capture and distinguish the different semantic information between these triples.

The experimental results in Tables 4 and 5 show that our model can improve the extraction of overlapping relations by applying relational hierarchy obtained by manual labeling or unsupervised clustering, across different overlapping patterns and different amounts of overlap. Thus, the experiments show that the relational hierarchy we constructed contains the necessary knowledge for extracting overlapping relational triples.

4.5.4. Influence of different ways for acquiring relational hierarchy on model results

The construction of the relationship hierarchy in this model can be either manually labeled or obtained by unsupervised clustering. When obtained by clustering, the different mapping functions of the base-level and higher-level relationships can affect the representations of each level relations, and thus the clustering results which can lead to different final results. The following is an analysis of the influence of the different mapping functions of the base-level and higher-level relationships on the model results.

For the WebNLG dataset without manually annotated hierarchical relations, higher-level relation sets can only be obtained by clustering automatically. The mapping functions for obtaining the representation of the base-level relations are "average" and "sum", while the mapping functions for getting the representation of higher-level relations are "average", "maximum," and "sum". Therefore, there are six different combinations of mapping functions for base-level and high-level relations. For example, the first "Avg" in "AvgAvg" is the mapping function of the base-level relation, and the second "Avg" is that of the higher-level relation.

For the NYT dataset with a manually annotated relational structure, there are two approaches to obtain the relational hierarchy, automatic clustering, and manual annotation. "Average" and "sum" mapping functions are used to obtain the representation of the base-level relations on

https://doi.org/10.1017/S1351324923000050 Published online by Cambridge University Press

Table 6. Different results with different mapping functions on validation set for representing base-level and higher-level relations in two datasets. The highest score of the model in different datasets is marked in bold when using clustering or not. "w/ category prefix" indicates that at the base-level relationship clustering, the relationship name is incorporating category labels; "w/o category prefix" indicates that the relationship name is not incorporating category labels

			w/o category prefix			w/ category prefix		
Dataset	Clustering	Pooling method	Prec.	Rec.	F1	Prec.	Rec.	F1
WebNLG	yes	AvgAvg	0.9198	0.9272	0.9235	–	–	–
		AvgMax	0.9165	0.9272	0.9218	–	–	–
		AvgSum	0.9244	0.9227	0.9236	–	–	–
		SumAvg	**0.9225**	**0.9308**	**0.9267**	–	–	–
		SumMax	0.9243	0.9218	0.9231	–	–	–
		SumSum	0.9073	0.9326	0.9198	–	–	–
NYT	yes	**AvgAvg**	0.9086	0.8973	0.9029	**0.9091**	**0.9038**	**0.9065**
		AvgMax	0.9061	0.9018	0.9040	0.8997	0.9076	0.9036
		AvgSum	0.9026	0.9043	0.9035	0.9066	0.8996	0.9031
		SumAvg	**0.8987**	**0.9106**	**0.9046**	0.8955	0.9134	0.9044
		SumMax	0.8962	0.9094	0.9027	0.9163	0.8961	0.9061
		SumSum	0.8999	0.9066	0.9033	0.8987	0.9105	0.9045
	no	AvgAvg	–	–	–	0.9078	0.9073	0.9076
		AvgSum	–	–	–	0.9078	0.9045	0.9061
		SumAvg	–	–	–	**0.9054**	**0.9131**	**0.9092**
		SumSum	–	–	–	0.9059	0.9041	0.9050

both approaches. The mapping functions of higher-level relations are the same as that of the WebNLG dataset when employing automatic clustering approach to obtain the hierarchy structure. However, the mapping functions of the higher-level relations are "average" and "sum," when using the manually annotated hierarchy.

Also, since the relations in the NYT dataset contain category labels, such as the name of relation **/people/person/nationality** contains the category labels **/people/person**. To determine whether category labels have an impact on the final results when constructing the relational hierarchy, the names of relations are constructed in two ways at the base-level relationship clustering, with category labels, that is, using **/person/person/nationality** as the relationship name, or without category labels, that is, using **nationality** as the relationship name.

It is observed in Table 6, that when the mapping functions of relations are "sum" or "average," the model easily obtains a better result. The relationship names in the WebNLG dataset do not contain category labels, so when using the unsupervised clustering method to obtain the relational hierarchy, only the "w/o category prefix" results are available, and the model results are better when the mapping function is "SumAvg." The relationship names in the NYT dataset contain category labels, so when using the manually annotated hierarchy, the model works better with the mapping function "SumAvg." When using the clustering method to obtain the relational hierarchy, the model with the mapping function "AvgAvg" worked better if the names of relations

https://doi.org/10.1017/S1351324923000050 Published online by Cambridge University Press

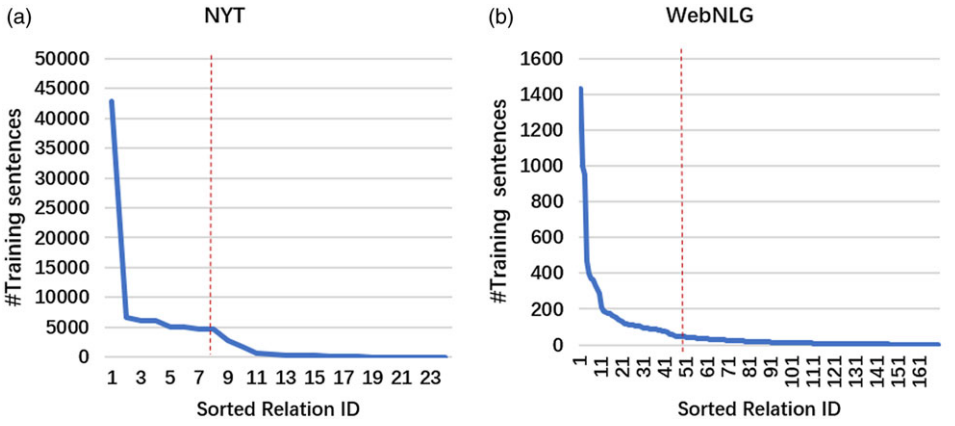

Figure 7. Label frequency distribution of positive relations in NYT and WebNLG datasets.

contained category labels; otherwise, the model with the mapping function "SumAvg" worked better. The reason why the mapping function works better when it consists of "sum" or "average" may be that "sum" or "average" has the potential to obtain more comprehensive information about the relationship, while "maximum" ignores the small but useful information about the relationship.

We can also observe in Table 6 that in the NYT dataset, when constructing the relational hierarchy, there is little difference in the results of the model between clustering and manual annotation. And, when constructing relational hierarchy by unsupervised clustering, the final results with or without the inclusion of category labels in the relationship names show that the incorporation of category labels has little effect on the final results.

In addition, we also analyze the effect of whether or not to incorporate category labels on the clustering results. See Appendix for details.

4.5.5. Influence of integrating hierarchical relational information on long-tail relation extraction

In this subsection, we analyze the effectiveness of fusing hierarchical relationship information in extracting long-tail relationships. Figure 7 depicts the label frequency distribution of positive relations in the NYT and WebNLG datasets. It shows that each dataset has approximately 70% of the relationships that occur very infrequently in the sentences. This problem seriously disrupts the balance of the data. To validate the effectiveness of our model and the attention mechanism in each level for extracting long-tail relationships, we use a subset of the extracted test dataset to test the model, in which the training instances of all the relations are less than 100/200 (Han *et al.* 2018).

We report average F1 values, micro-F1 values, and macro-F1 values for the long-tail relationships in the dataset. Since the average microscore typically ignores the effect of long-tail relationships, we use average macroscores to highlight long-tail relationships in the test set, which has often been overlooked in previous work. Table 7 presents the detailed results comparing the scores with the state-of-the-art model where the number of relations in the test dataset is less than 100/200 instances in the training dataset.

From Table 7, we observed that our model employing the attention mechanism using relational hierarchy does improve the extraction of long-tail relationships over the $Casrel_{BERT}$ model that does not use it. Our model is better for the extraction of long-tail relationships than using only the local attention mechanism, or only the global attention mechanism, to obtain the relationship information, where the enhancement effect of the relationship information obtained through the global attention mechanism is greater than that of information obtained through the local attention mechanism for long-tail relationship extraction. This indicates that the use of coarse-grained

https://doi.org/10.1017/S1351324923000050 Published online by Cambridge University Press

Table 7. Mean-F1, Micro-F1, and Macro-F1 using Partial entity matching between our model and the state-of-the-art model on relations with fewer than 100/200 training instances

		<100			<200		
Dataset	Model	Micro-F1	Macro-F1	Mean-F1	Micro-F1	Macro-F1	Mean-F1
WebNLG	$CasRel_{BERT}$	92.68	88.92	86.71	92.45	89.39	87.52
	$Our_{Cluster}$	**93.63**	**90.96**	**89.43**	**93.52**	**91.34**	**90.06**
	–local	93.36	89.10	87.62	93.15	89.69	88.45
	–global	92.37	89.07	87.40	92.02	89.38	87.97
NYT	$CasRel_{BERT}$	89.65	90.90	89.33	89.26	90.63	89.29
	$Our_{Cluster}$	**93.33**	**93.80**	**93.14**	**95.34**	**94.41**	**93.89**
	–local	85.71	81.18	79.80	93.02	85.63	83.99
	–global	85.71	81.18	79.80	90.69	84.82	83.19
	Our_{Manual}	**93.33**	**93.80**	**93.14**	**91.56**	**93.05**	**92.58**
	–local	90.32	91.54	91.00	90.39	91.26	90.83
	–global	87.50	87.49	85.14	89.88	88.32	86.64

and fine-grained semantic information can improve the extraction of long-tail relationships, and the coarse-grained semantic information is more useful for the extraction of overlapping triples of long-tailed relations.

We also found that for the NYT dataset, whether using clustering to obtain hierarchies or manually labeled hierarchies, our model employing the attention mechanism using relational hierarchy showed almost no change in the results for relationships with less than 100 instances in the training set, but for relationships with less than 200 instances in the training set, clustering to obtain relational hierarchies were extracted significantly better than manually labeled hierarchies. However, the results in Table 2 show that the hierarchy obtained by clustering is not as effective as that obtained by manual labeling in the NYT dataset, indicating that the clustering error has more influence on the extraction effect of the head relations that occur more frequently, but for the relations with instances greater than 100 and less than 200 in the training set, multi-granularity relationship knowledge provides more useful information.

5. Conclusion

This paper proposes an end-to-end neural framework that merges the multi-granularity relational features for overlapping triple extractions. We employ an attention mechanism that uses a relational hierarchy to capture the coarse-to-fine semantic information hidden in multiple overlapping relational triples. The relational hierarchy can be obtained by using manual annotation or automatic clustering.

We evaluated our model on the NYT and WebNLG datasets and conducted various experiments. Experimental results show that our model outperforms previous work in extracting overlapping relational triples, where the relational hierarchies obtained by automatic clustering are not much worse than those obtained by manual annotation, and the integration of multi-granularity relational knowledge is indeed effective in improving the extraction of long-tailed data.

https://doi.org/10.1017/S1351324923000050 Published online by Cambridge University Press

In the future, we plan to construct three levels of relationship hierarchies to compare the impact on different construction methods of the results and also consider ways to reduce the information redundancy after incorporating the multi-granularity relationship features of the head relationships.

Acknowledgments. The research reported in this paper was supported in part by the Shanghai Municipal Science and Technology Committee of Shanghai Outstanding Academic Leaders Plan 20XD1401700; National Key Research and Development Program of China 2021YFC3300602; the Natural Science Foundation of China under the grant No.91746203; the National Natural Science Foundation of China under the grant No.61991415; and Shanghai Science and Technology Young Talents Sailing Program Grant 21YF1413900.

References

Bekoulis G., Deleu J., Demeester T. and Develder C. (2018). Joint entity recognition and relation extraction as a multi-head selection problem. *Expert Systems with Applications* **114**, 34–45.

Bi W. and Kwok J.T. (2011). Multilabel classification on tree-and dag-structured hierarchies. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pp. 17–24.

Chen J., Yuan C., Wang X. and Bai Z. (2019). Mrmep: Joint extraction of multiple relations and multiple entity pairs based on triplet attention. In *Proceedings of the 23rd Conference on Computational Natural Language Learning, CoNLL 2019, Hong Kong, China, November 3-4, 2019*, pp. 593–602.

Devlin J., Chang M.-W., Lee K. and Toutanova K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. *Association for Computational Linguistics*, pp. 4171–4186.

Frey B.J. and Dueck, D. (2007). Clustering by passing messages between data points. *Science* **315**, 972–976.

Fu T.-J., Li P.-H. and Ma W.-Y. (2019). Graphrel: Modeling text as relational graphs for joint entity and relation extraction. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1409–1418.

Gardent C., Shimorina A., Narayan S. and Perez-Beltrachini L. (2017). Creating training corpora for NLG micro-planning. In *55th Annual Meeting of the Association for Computational Linguistics* (ACL), pp. 179–188.

Gormley R.M., Yu M. and Dredze M. (2015). Improved relation extraction with feature-rich compositional embedding models. In *Conference on Empirical Methods* in Natural Language Processing, pp. 1774–1784.

Gupta P., Schütze H. and Andrassy B. (2016). Table filling multi-task recurrent neural network for joint entity and relation extraction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 2537–2547.

Han X., Yu P., Liu Z., Sun M. and Li P. (2018). Hierarchical relation extraction with coarse-to-fine grained attention. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2236–2245.

Hu Z., Huang P., Deng Y., Gao Y. and Xing P.E. (2015). Entity hierarchy embedding. *The Association for Computer Linguistics*, pp. 1292–1300.

Kai S., Richong Z., Samuel M., Yongyi M. and Xudong L. (2020). Recurrent interaction network for jointly extracting entities and classifying relations. In *EMNLP* 2020, pp. 3722–3732.

Li Q. and Ji H. (2014). Incremental joint extraction of entity mentions and relations. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 402–412.

Li Y., Shen T., Long G., Jiang J., Zhou T. and Zhang C. (2020). Improving long-tail relation extraction with collaborating relation-augmented attention. *COLING*, pp. 1653–1664.

Liang T., Liu Y., Liu X., Sharma G. and Guo M. (2021). Distantly-supervised long-tailed relation extraction using constraint graphs. *IEEE Transactions on Knowledge and Data Engineering*, 1–1.

McCallum A., Rosenfeld R., Mitchell T.M. and Ng A.Y. (1998). Improving text classification by shrinkage in a hierarchy of classes. *ICML* **98**, 359–367.

Mintz M., Bills S., Snow R. and Jurafsky D. (2009). Distant supervision for relation extraction without labeled data. *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*, pp. 1003–1011.

Miwa M. and Sasaki Y. (2014). Modeling joint entity and relation extraction with table representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1858–1869.

Nayak T. and Ng H.T. (2020). Effective modeling of encoder-decoder architecture for joint entity and relation extraction. *Proceedings of the AAAI Conference on Artificial Intelligence* **34**, 8528–8535.

Ren X., Wu Z., He W., Qu M., Voss C.R., Ji H., Abdelzaher T.F. and Han J. (2017). Cotype: Joint extraction of typed entities and relations with knowledge bases. *Proceedings of the 26th International Conference on World Wide Web*, pp. 1015–1024.

Riedel S., Yao L. and McCallum A. (2010). Modeling relations and their mentions without labeled text. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 148–163.

Rousu J., Saunders C., Szedmak S. and Shawe-Taylor J. (2005). Learning hierarchical multi-category text classification models. In *Proceedings of the 22nd International Conference on Machine Learning*, pp. 744–751.

Takanobu R., Zhang T., Liu J. and Huang, M. (2019). A hierarchical framework for relation extraction with reinforcement learning. In *International Conference on Artificial Intelligence*, pp. 7072–7079.

Tipping M.E. and Bishop C.M. (1999). Mixtures of probabilistic principal component analysers. *Neural Comput* **11**, 443–482.

Verma N., Mahajan D., Sellamanickam S. and Nair V. (2012). Learning hierarchical similarity metrics. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2280–2287.

Wang Y., Yu B., Zhang Y., Liu T., Zhu H. and Sun L. (2020). Tplinker: Single-stage joint extraction of entities and relations through token pair linking. In *COLING*, pp. 1572–1582.

Wei Z., Su J., Wang Y., Tian Y. and Chang Y. (2020). A novel cascade binary tagging framework for relational triple extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pp. 1476–1488.

Weinberger K.Q. and Chapelle O. (2009). Large margin taxonomy embedding for document categorization. In *Advances in Neural Information Processing Systems*, pp. 1737–1744.

Xiao L., Zhou D. and Wu M. (2011). Hierarchical classification via orthogonal transfer. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pp. 801–808.

Xie R., Liu Z. and Sun M. (2016). Representation learning of knowledge graphs with hierarchical types. *IJCAI*, 2965–2971.

Ye H., Zhang N., Deng S., Chen M., Tan C., Huang F. and Chen H. (2021). Contrastive triple extraction with generative transformer. *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pp. 14257–14265.

Yu B., Zhang Z., Shu X., Wang Y., Liu T., Wang B. and Li S. (2020). Joint extraction of entities and relations based on a novel decomposition strategy. In *ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020)*, 325:2282–2289.

Yu X. and Lam W. (2010). Jointly identifying entities and extracting relations in encyclopedia text via a graphical model approach. *Coling 2010: Posters*, pp. 1399–1407.

Yuan Y., Zhou X., Pan S., Zhu Q., Song Z. and Guo L. (2020). A relation-specific attention network for joint entity and relation extraction. *IJCAI 2020*, pp. 4054–4060.

Zeng D., Zhang H. and Liu Q. (2020). Copymtl: Copy mechanism for joint extraction of entities and relations with multi-task learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9507–9514.

Zeng X., He S., Zeng D., Liu K., Liu S. and Zhao J. (2019). Learning the extraction order of multiple relational facts in a sentence with reinforcement learning. 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)*, pp. 367–377.

Zeng X., Zeng D., He S., Liu K. and Zhao J. (2018). Extracting relational facts by an end-to-end neural model with copy mechanism. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 506–514.

Zhang N., Deng S., Sun Z., Wang G., Chen X., Zhang W. and Chen H. (2019). Long-tail relation extraction via knowledge graph embeddings and graph convolution networks. *Association for Computational Linguistics*, 3016–3025.

Zhang N., Deng S., Ye H., Zhang W. and Chen H. (2022). Robust triple extraction with cascade bidirectional capsule network. *Expert Systems With Applications*, **187**, 115806.

Zhang N., Ye H., Deng S., Tan C., Chen M., Huang S., Huang F. and Chen H. (2021). Contrastive information extraction with generative transformer. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 3077–3088.

Zhang Z., Zhuang F., Qu M., Lin F. and He Q. (2018). Knowledge graph embedding with hierarchical relation structure. In *EMNLP*, 3198–3207.

Zhao B., Li F.-F. and Xing P.E. (2011). Large-scale category structure aware image categorization. In *NIPS*, 1251–1259.

Zheng H., Wen R., Chen X., Yang Y., Zhang Y., Zhang Z., Zhang N., Qin B., Xu M. and Zheng Y. (2021). PRGC: Potential relation and global correspondence based joint relational triple extraction. In *Annual Meeting of the Association for Computational Linguistics*, 6225–6235.

Zheng S., Wang F., Bao H., Hao Y., Zhou P. and Xu B. (2017). Joint extraction of entities and relations based on a novel tagging scheme. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pp. 1227–1236.

Appendix

Influence of removing category prefixes on the clustering results

In this section, we also analyze how removing the category prefixes affects the clustering results. As seen in Figure 8, removing category labels makes the clustering results different, even if the mapping functions of the base-level relations are the same. For example, when using "average"

https://doi.org/10.1017/S1351324923000050 Published online by Cambridge University Press

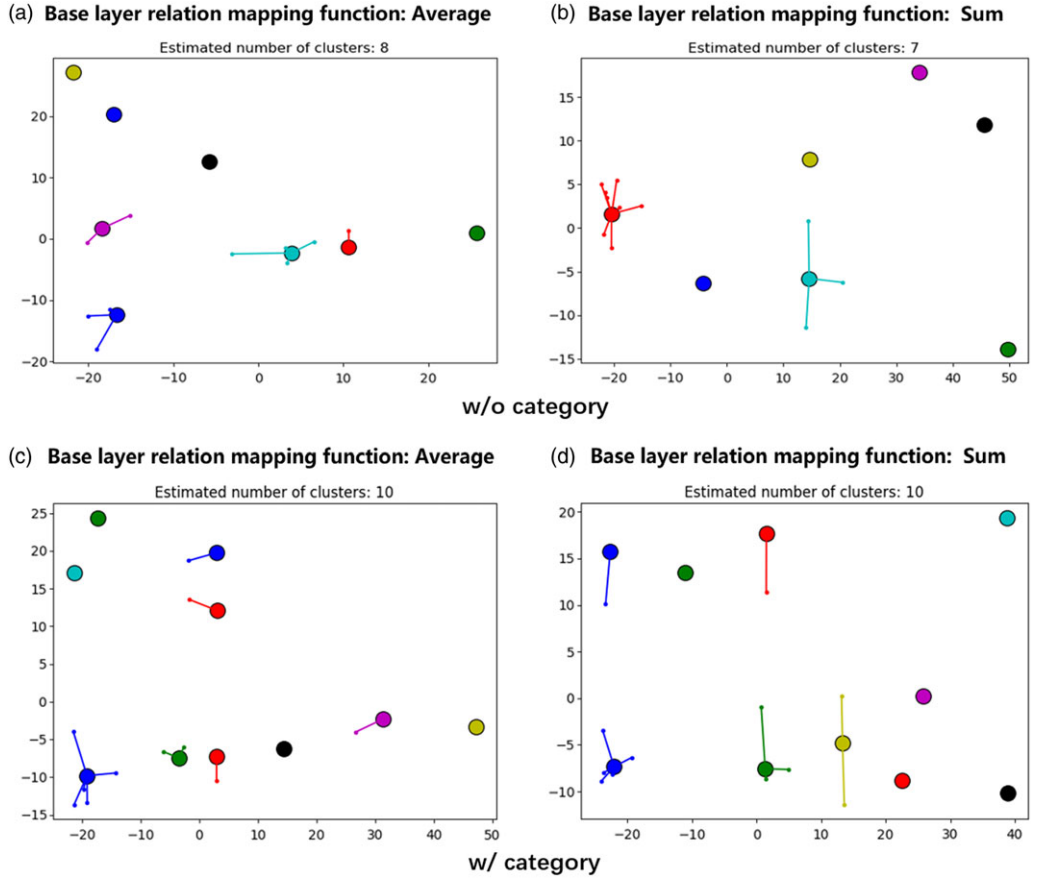

Figure 8. Clustering results for different mapping functions of base-level relationships in the NYT dataset. (a) and (c) are the clustering results when the mapping functions of base-level relations are "average", while (b) and (d) are the clustering results when the mapping functions of base-level relations are "sum". "w/ category prefix" indicates that at the base-level relationship clustering, the relationship name is incorporating category labels; "w/o category prefix" indicates that the relationship name is not incorporating category labels.

as the mapping function of the base-level relations, the number of clusters is 10 if the names of relations contain category labels and 8 if the names of relations do not contain category labels. In Table 8, we analyze the effect of category labels on the clustering results, we collated the classification results for cluster 0 and cluster 1 when "average" was used as the mapping function for the base-level relationship.

We can clearly find that removing the category labels from the relationship names affects the classification results to some extent which is shown by the fact that the classification results of clusters are somewhat different from the manually labeled ones.

Thus, Table 8 and the above results show that although removing the category label of the names of relations can affect the classification results, the change in classification results has a weak influence on the final results of the model. Therefore, in datasets where there is no manually labeled hierarchy, the relational hierarchy can be obtained using unsupervised clustering.

https://doi.org/10.1017/S1351324923000050 Published online by Cambridge University Press

Table 8. Samples of clustering results including the cluster 0 and the cluster 1 in the NYT dataset when the base-level relations using the "average" mapping function. "w/ category prefix" indicates that we cluster relation names with using explicit category prefixes; "w/o category prefix" indicates that we cluster relation names without using explicit category prefixes.

Clustering results	w/ category prefix	w/o category prefix
cluster 0	/business/company/advisors /business/company/industry /business/company/major_shareholders	/business/company/founders /business/person/company /people/ethnicity/people /people/person/profession
cluster 1	/business/company/founders	/business/company/advisors /business/company/industry /location/administrative_division/country /location/location/contains /sports/sports_team/location

Cite this article: Su H, Wang H, Luo X and Xie S (2023). An end-to-end neural framework using coarse-to-fine-grained attention for overlapping relational triple extraction. *Natural Language Engineering* **29**, 1126–1149. https://doi.org/10.1017/S1351324923000050