



# Attributed Triple Extraction by Combination Under Contrastive Learning

Runzhe Wang<sup>1,2</sup>, Gaofeng Xu<sup>1,2</sup>, Guilin Qi<sup>1,2</sup>(✉), Yun Li<sup>3</sup>, Yongrui Chen<sup>1,2</sup>,  
Songlin Zhai<sup>1,2</sup>, Yuxin Zhang<sup>1,2</sup>, Rihui Jin<sup>1,2</sup>, Nijun Li<sup>3</sup>, and Qianren Wang<sup>3</sup>

<sup>1</sup> School of Computer Science and Engineering, Southeast University,  
Nanjing 211189, China

{220232335, 230229361, gqi, yrchen, songlin\_zhai, zzyx\_cs, 220212061}@seu.edu.cn

<sup>2</sup> Key Laboratory of New Generation Artificial Intelligence Technology and its  
Interdisciplinary Applications (Southeast University), Ministry of Education,  
Beijing, China

<sup>3</sup> Advanced Cognitive AI Lab, Huawei Technologies, Shenzhen, China  
{liyun5, linijun, wangqianran}@huawei.com

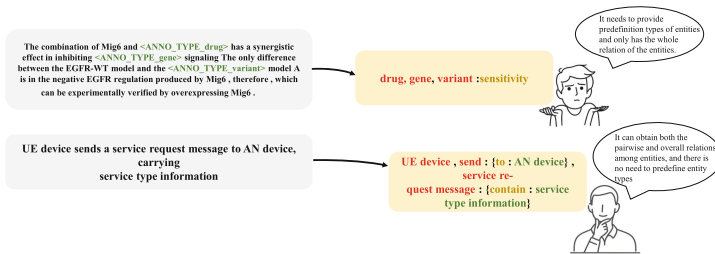
**Abstract.** Complex knowledge extraction is a significant topic of research. Previous methodologies were limited to extracting only the overall relations among multiple entities and they necessitated the predefined entity types, rendering them inapplicable in industrial domains such as Information and Communication Technology (ICT), where numerous entities are undefined. To address these challenges, this paper introduces a novel representation for complex knowledge: **attributed triple**, which consists of a primary triple coupled with auxiliary attribute-value pairs, enabling a fine-grained depiction of both the overall inter-entity relations and the pairwise relations between entities. Based on this representation, we propose an attributed triple extraction model. This model employs multi-label classifiers to identify all potential relations before extracting relevant entities with sequence taggers based on the **BIO** pattern, where **B** represents **B**egin, **I** represents **I**nside and **O** represents **O**utside. This process makes the extraction of primary triples and auxiliary pairs without predefining entity types. Subsequently, a linear combination discriminator is designed to assess the semantic feasibility of candidate combinations formed by concatenating primary triplets and auxiliary pairs. Furthermore, contrastive learning is adopted to enhance the model's ability of representation with insufficient training data. To better evaluate the model's performance over extracting attributed triple, we construct a high-quality dataset based on data from ICT corpus. Our model demonstrates substantial and consistent superiority over baselines across various metrics. Our code is publicly available. <https://github.com/sid0527/Attributed-Triple-Extraction>

**Keywords:** Attributed Triple · Information Extraction · Contrastive Learning

---

The author contributed equally to this work.

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2024  
M. Onizuka et al. (Eds.): DASFAA 2024, LNCS 14856, pp. 402–414, 2024.  
[https://doi.org/10.1007/978-981-97-5575-2\\_30](https://doi.org/10.1007/978-981-97-5575-2_30)



**Fig. 1.** The differences in the various representations of complex data. In the figure, the red font indicates entities, green indicates entity types and yellow indicates relations. (Color figure online)

## 1 Introduction

Information extraction, an essential process in the construction and completion of knowledge graphs [1, 2], facilitates the automated retrieval of structured information from voluminous text. Previous works mainly focused on binary relational information extraction, which is also called triple extraction [3]. However, in numerous industrial scenarios, the text encapsulates complex knowledge characterized by interactions among multiple entities. The extraction of such intricate knowledge is significant as it can enrich knowledge graphs with more comprehensive and detailed entity-relation information.

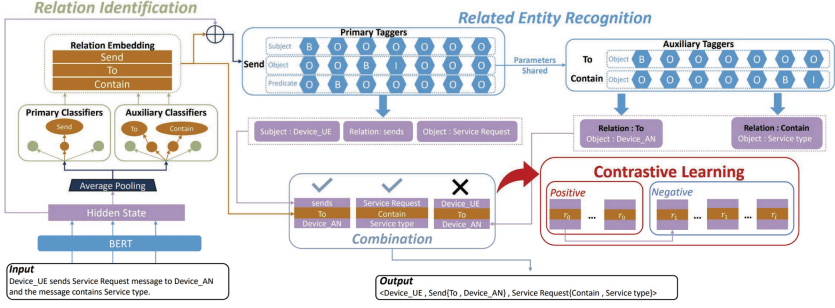
Current studies on complex knowledge extraction lie primarily within the biomedical domain, which involves distilling an overarching relation among specific categories of entities [4–7]. These methods necessitate the predefinition of entity types prior to extraction, as depicted in Fig. 1. However, their application in other industrial fields, such as Information and Communication Technology (ICT), can lead to two major challenges. 1) Beyond the overall relations among entities, individual relations between entities also matter. For instance, **UE device sends Service Request message to AN device, and the message contains Service type**, where **UE** and **AN** are undefined entities in ICT. Here, the relation between **UE device** and **Service Request message** is defined as **send**, while the relation between **Service Request message** and **Service type** is **contain** and the collective relation of the entities (**UE device**, **Service Request message**, **AN device**) is (**send...to...**). Representing these entities using a single overarching relation can compromise information integrity. 2) The diversity and continual emergence of new entity types in textual data make the provision of entity-type predefinition impractical.

Intuitively, the ideal representation would involve a more granular form of complex knowledge, capable of articulating both the pairwise and overall relations among multiple entities. Consequently, we distill complex text into primary components and modifiers, achieving a fine-grained representation of complex knowledge. Moreover, we observe that the range of relations in most extraction

datasets is considerably narrower than the variety of entity types. An intuitive solution is to identify all relevant entity types within the text by recognizing relations, thus circumventing the need of entity-type predefinition.

In this paper, we introduce a novel representational form for complex knowledge: **attributed triple**, comprising a primary triple allied with auxiliary attribute-value pairs. The primary triple delineates the central subject-predicate-object structure of the text, while the auxiliary pairs provide additional contextual details. The instance mentioned above can be represented as (UE device, send:{to, AN device}, Service Request message:{contain, Service type}), where (UE device, send, Service Request message) constitutes the primary triple, and {to, AN device}, {contain, Service type} are auxiliary pairs. Based on this representation, we propose an attributed triple extraction model. This model employs multi-label classifiers to identify potential relations, followed by entity extraction through a sequence tagger based on the BIO pattern. This process yields the primary triple and auxiliary pairs without predefining entity types. A linear combination discriminator is then used to evaluate the semantic feasibility of candidate combinations, formed by concatenating primary triplets and auxiliary pairs. To reduce the effect of industrial data scarcity on model's ability of representation, contrastive learning is integrated during the training phase. Moreover, to assess the model's effectiveness in extracting attributed triples, we have developed a high-quality dataset derived from the ICT domain. This dataset, featuring a variety of primary and auxiliary relations with diverse combinations, more accurately mirrors the complexities of real-world industrial scenarios, in contrast to existing data. Extensive experimental results demonstrate that our model can effectively extract high-quality attributed triples, proving to be of significant value in practical applications. In summary, the contributions of this paper are threefold:

1. We introduce a new representational form for multi-entity information: **attributed triple**, which is capable of depicting both the pairwise and overall relations among entities, thereby enabling fine-grained representation of information among multiple entities.
2. We have developed an attributed triple extraction model that prioritizes relation identification over entity recognition, negating the need for predefining entity type. This model integrates contrastive learning and employs a shared loss function during training, enhancing the accuracy of the model outputs and substantially improving its practicality.
3. To thoroughly evaluate the model's capacity for extracting attributed triples, we semi-automatically curated a high-quality dataset based on the ICT domain. This dataset comprises a diverse array of primary and auxiliary relations, better mirroring actual industry scenarios compared to open datasets. Extensive experiments on both open and industrial datasets demonstrate that our model achieves excellent performance across a range of metrics.



**Fig. 2.** An overview of **Combination under Contrastive Learning**. Both relation identification and related entity recognition module are designed to be composed of two similar components, which benefits the separate extraction of primary triples and auxiliary relation-entity pairs. To address insufficiency of training data, parameter sharing mechanism and contrastive learning are introduced.

## 2 Preliminaries

### 2.1 Attributed Triple

An attributed triple  $\mathcal{F}$  is a primary subject-relation-object triple  $(s, r, o)$  coupled with  $m$  auxiliary pairs  $\{(a_i, v_i)\}_{i=1}^m$ , where  $r, a_1, \dots, a_m \in \mathcal{R}$  and  $s, o, v_1, \dots, v_m \in \mathcal{E}$ , with  $\mathcal{R}$  and  $\mathcal{E}$  being the sets of relations and entities, respectively. We denote the attributed triple as  $\mathcal{F} = (s : \{(a_i, v_i)\}_{i=1}^{m_s}, r : \{(a_i, v_i)\}_{i=1}^{m_r}, o : \{(a_i, v_i)\}_{i=1}^{m_o})$ , where  $m_s + m_r + m_o = m$  and  $x : y$  denotes  $y$  attribute to  $x$ . We slightly abuse terminology here by referring to the primary relation and auxiliary attribute as relations, and referring to the subject, object, and values as entities unless otherwise specified.

### 2.2 Problem Formulation

Our goal is to extract attributed triples from text. Given the input text  $\mathcal{T} = \{t_1, t_2, \dots, t_n\}$ , we aim to extract  $\mathcal{G}(\mathcal{T}) \rightarrow \mathcal{F}$ , where  $\mathcal{G}$  denotes the attributed triples extraction model.

## 3 Combination Under Contrastive Learning

This section presents CCL, our model to extract attributed triples by combination under contrastive learning. There are two key modules in our model: attributed triple combination and contrastive learning. The former combines attributed triples by **Relation Identification**, **Related Entity Recognition** and **Combination**, and the latter designs a loss to address data scarcity. An overview illustration of our model is shown in Fig. 2.

### 3.1 Attributed Triple Combination

A pre-trained BERT is employed as encoder to transform the input text  $S$  into the  $d$ -dimensional contextual representation:

$$[\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_L] = \text{Encoder}(\mathbf{S}), \quad (1)$$

where  $\mathbf{h}_i$  corresponds to the  $i$ -th input token and  $L$  is the length of  $S$ .

**Relation Identification.** Different from works [8, 9] which extract all candidate entities before permutating them to form triples, we first obtain subsets of potential primary relation types and auxiliary relation types respectively, and only perform entity extraction for these potential relations. Since much more possible entity combinations is taken into consideration in one attributed triple, this part will greatly improve the efficiency of extraction.

With the output of the encoder  $\mathbf{H}_{enc} = [\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_L]$ , probability of one relation type being potential in text  $\mathbf{S}$  can be predicted by:

$$P_{rel} = \text{Sigmoid}(\mathbf{W}_{rel} \mathbf{h}_{enc}^{avg} + \mathbf{b}_{rel}), \quad (2)$$

where  $\mathbf{h}_{enc}^{avg}$  is the average vector among all  $\mathbf{h}_i$  in  $\mathbf{H}_{enc}$ .  $\mathbf{W}_{rel}$  and  $\mathbf{b}_{rel}$  represent a trainable weight and a trainable bias, respectively. The subset of potential relations can be obtained as  $\mathbf{R}_{pot} = \{r_i | P_{r_i} > \lambda\}$ , where  $\lambda$  is a preset threshold.

According to cross-entropy loss, loss function of this part can be defined as:

$$l_{rel}^{pri} = -\frac{1}{n_{rel}^{pri}} \sum_{i=1}^{n_{rel}^{pri}} \left( y_i^{pri} \log(P_{rel}) + (1 - y_i^{pri}) \log(1 - P_{rel}) \right), \quad (3)$$

$$l_{rel}^{aux} = -\frac{1}{n_{rel}^{aux}} \sum_{i=1}^{n_{rel}^{aux}} \left( y_i^{aux} \log(P_{rel}) + (1 - y_i^{aux}) \log(1 - P_{rel}) \right), \quad (4)$$

where  $n_{rel}^{pri}$  and  $n_{rel}^{aux}$  denote the size of primary and auxiliary relation set.

**Related Entity Recognition.** Potential primary relation types and auxiliary relation types obtained will be the input of this part in the form of embedding, together with that of each token in text  $\mathbf{S}$  [10]. We can update the representation of  $i$ -th token in the context of potential relation type  $r_j$  to:

$$\mathbf{h}_{i,j} = \mathbf{h}_i + \mathbf{v}_{r_j}, \quad (5)$$

where  $\mathbf{v}_j \in \mathbb{R}^{d \times 1}$  is a trainable embedding vector for the  $j$ -th relation. Related entity recognition is based on the new representation through sequence tagging in the BIO pattern as is shown in Fig. 2.

When it comes to the network in this component, we dismiss the LSTM-CRF [11], which proves to be effective but may be too complicated to be trained well in consideration of the insufficiency of training data in some industrial domains.

Instead, FCNN is adopted to simplify the network. For each potential primary relation type, there are three entities to be recognized: subject, object, and relation. Hence, three rounds of sequence tagging will be performed independently:

$$P_{i,j}^{sub} = \text{Softmax}(\mathbf{W}_{sub}\mathbf{h}_{i,j} + \mathbf{b}_{sub}), \quad (6)$$

$$P_{i,j}^{obj} = \text{Softmax}(\mathbf{W}_{obj}\mathbf{h}_{i,j} + \mathbf{b}_{obj}), \quad (7)$$

$$P_{i,j}^{re} = \text{Softmax}(\mathbf{W}_{re}\mathbf{h}_{i,j} + \mathbf{b}_{re}), \quad (8)$$

where  $\mathbf{W}_{(\cdot)} \in \mathbb{R}^{3 \times d}$  is a trainable weight and  $\mathbf{b}_{(\cdot)} \in \mathbb{R}^{3 \times 1}$  is a trainable bias. For each potential auxiliary relation type, we just take advantage of  $\mathbf{W}_{obj}$  and  $\mathbf{b}_{obj}$  to get the object in the context of this relation.

Loss function of three taggers can be calculated by:

$$l_{seq}^{sub} = - \sum_{j=1}^{n_{pre}^{pri}} \sum_{i=1}^L \frac{y_{i,j}^{sub} \log(P_{i,j}^{sub})}{L \times n_{pre}^{pri}}, \quad (9)$$

$$l_{seq}^{men} = - \sum_{j=1}^{n_{pre}^{pri}} \sum_{i=1}^L \frac{y_{i,j}^{pre} \log(P_{i,j}^{pre})}{L \times n_{pre}^{pri}}, \quad (10)$$

$$l_{seq}^{obj} = - \frac{1}{2 \times L} \left[ \sum_{j=1}^{n_{pre}^{pri}} \sum_{i=1}^L \frac{y_{i,j}^{obj} \log(P_{i,j}^{obj})}{n_{pre}^{pri}} - \sum_{j=1}^{n_{pre}^{aux}} \sum_{i=1}^L \frac{y_{i,j}^{obj} \log(P_{i,j}^{obj})}{n_{pre}^{aux}} \right], \quad (11)$$

where  $L$  is the length of input text.  $n_{pre}^{pri}$  and  $n_{pre}^{aux}$  represent the size of potential primary relation type set and that of auxiliary relation type set.

**Combination.** Each auxiliary pair will be concatenated with all primary triplets respectively to generate candidate combinations. To determine whether a combination is semantically valid, a linear classifier is introduced in this component.

It is worth mentioning that each part of the primary triple is an entity composed of multiple tokens, which may hinder parallel computation. Therefore, we represent entities with an average vector to keep the dimension. The representation of every combination can be obtained as follows:

$$\mathbf{h}_{i,j}^{com} = [\mathbf{h}_{e_i}; \mathbf{v}_{r_j}; \mathbf{h}_{e_j}], \quad (12)$$

where  $e_i$  is an entity in one primary triple and  $e_j$  is the object corresponding to auxiliary relation type  $r_j$ . Symbol  $;$  represents the operation of concatenation. Probability of such combination being valid can be measured by:

$$P_{i,j}^{com} = \text{Sigmoid}(\mathbf{W}_{com}\mathbf{h}_{i,j}^{com} + \mathbf{b}_{com}), \quad (13)$$

where  $\mathbf{W}_{com}$  is a trainable weight and  $\mathbf{b}_{com}$  is a trainable bias. If  $P_{i,j}^{com}$  is the greatest one among all candidate combinations containing the auxiliary pair  $(r_j, e_j)$ ,  $(r_j, e_j)$  will be combined with  $e_i$  into an attributed triple.

The loss of combination is similar to that of relation identification:

$$l_{com} = - \sum_{i=1}^{n_{pri}} \sum_{j=1}^{n_{aux}} \left( \frac{y_{i,j} \log(P_{i,j}^{com}) + (1 - y_{i,j}) \log(1 - P_{i,j}^{com})}{n_{pri} \times n_{aux}} \right), \quad (14)$$

where  $n_{pri}$  refers to the number of entities in the primary triples and  $n_{aux}$  refers to the number of auxiliary pairs in input text.

### 3.2 Contrastive Learning

Contrastive learning is introduced to enable our model to learn subtle differences between samples, which improves its capability of representation when training data is insufficient. The contrastive loss is calculated based on all candidate combinations in one batch, where two combinations with same relation type will be taken as positive.

Following previous works [12,13], we define the loss for a positive pair as:

$$l(\mathbf{h}^{com_i}, \mathbf{h}^{com_j}) = - \log \left( \frac{e^{(\mathbf{h}^{com_i} \cdot \mathbf{h}_j^{com} / \tau)}}{\sum_{k=0}^{n_{com}} \mathbb{1}_{i \neq k} \cdot e^{(\mathbf{h}^{com_i} \cdot \mathbf{h}^{com_k} / \tau)}} \right), \quad (15)$$

where  $com_i$  is representation of one combination,  $\tau$  is a temperature parameter, and  $n_{com}$  is number of candidate combinations in the batch.  $\mathbb{1}_{i \neq k}$  is the indicator function. The final loss is computed from all positive pairs in the batch.

### 3.3 Objective Function

We define the objective function as a total loss composed of loss from each module. With Eqs. 3 and 4, loss for relation identification is denoted as:

$$l_{rel} = \frac{1}{2} (l_{rel}^{pri} + l_{rel}^{aux}). \quad (16)$$

According to Eqs. 9, 10 and 11, the final loss for entity recognition is defined as an average of three loss values:

$$l_{seq} = \frac{1}{3} (l_{seq}^{sub} + l_{seq}^{obj} + l_{seq}^{men}). \quad (17)$$

The final objective function is the total of mentioned loss functions:

$$l = l_{rel} + l_{seq} + \alpha \times l_{com} + (1 - \alpha) \times l_{com}, \quad (18)$$

where  $0 \leq \alpha \leq 1$  is a weight to be tuned during training.

## 4 Experiment

### 4.1 Datasets

To evaluate the performance of our model on open data, we choose DuIE, a well-known dataset for information extraction, as the source of one dataset because some triples in DuIE contain complex objects. Each of these complex objects provides not only a object value but also some other slots that can be taken as auxiliary relations. In this way, we obtain a dataset with 2070 attributed triples in the newly proposed form.

Together with 3 experts, we also manually build a dataset based on information from ICT corpus. This dataset consists of 171 attributed triples, and it enables us to explore how our model behaves when applied to real industrial scenario. Details of the proposed datasets are shown in Table 1.

**Table 1.** Details of the proposed datasets.

Dataset	Composition	Relation type				
DuIE	Primary triple	Dub	Play	Award	Box Office	-
		461	1302	469	118	-
	Auxiliary pair	inWork	onDate	inArea	-	-
ICT		1978	340	118	-	-
	Primary triple	Support	Set	Send	Provide	-
		39	52	44	36	-
	Auxiliary pair	Be	TO	For	Contain	Through
		27	44	60	20	48

## 4.2 Experiment Settings

**Baseline.** In the absence of existing work for extracting our proposed complex relations, we employ a simplified version of our model as a baseline for comparison. This simplified model omits the contrastive learning and combination modules, using a straightforward nearest-neighbor principle for combination, effective in simpler contexts. Additionally, it alters the entity recognition module to train object recognizers independently for primary and auxiliary relations, without parameter sharing.

Given the proven excellence of large language models in various natural language processing tasks, we also utilize GPT-3.5-turbo as an additional baseline to assess our model’s performance.

**Metrics.** Our evaluation metrics for model performance include **Precision**, **Recall**, and **F1 score**, encompassing three aspects: primary triples, auxiliary pairs, and attributed triples. For primary triples and auxiliary pairs, we apply strict matching, deeming results correct only if they exactly match the annotations. In evaluating attributed triples, we combine strict and partial matching to more accurately gauge the model’s effectiveness. Equation 19 details the evaluation criteria for each attributed triple.

$$N_{precision} = \mathbb{1}_{primary\ triple} \cdot \frac{n_{correct}}{n_{total}}, N_{recall} = \mathbb{1}_{primary\ triple} \cdot \frac{n_{correct}}{n_{annotation}}, \quad (19)$$

where  $\mathbb{1}_{primary\ triple}$  equals to 1 only when the extracted primary triple is correct and  $n_{correct}$  is the number of auxiliary pairs extracted correctly.  $n_{total}$  and  $n_{annotation}$  represent the total number of auxiliary pairs extracted and the number of auxiliary pairs annotated respectively. The final **Precision** and **Recall** will be the average of all  $N_{precision}$  and  $N_{recall}$ .

**Details.** Our experiment, conducted on the PyTorch framework, utilized an NVIDIA RTX 3080Ti GPU and CUDA 11.3. We employed Hugging Face’s bert-chinese-base as the pre-trained BERT model, featuring a 768-dimension hidden text representation. The maximum sentence sequence length was set at 100. Batch sizes were 5 for training, 24 for validation, and 64 for testing, with a maximum of 40 training epochs. Hyperparameters were fine-tuned on the validation set, setting BERT’s learning rate at 1e-4 with a 0.01 weight decay, and 1e-3 for other model components.



**Table 2.** Precision(%), Recall (%) and F1-score (%) of CCL and baselines.

Dataset	Model	Composition	Prec.	Rec.	F1
DuIE	GPT-3.5-turbo	Primary	0.577	0.536	0.556
		Auxiliary	0.324	0.302	0.314
		Attributed	0.308	0.286	0.297
	Simplified CCL	Primary	0.468	0.552	0.507
		Auxiliary	0.213	0.235	0.223
		Attributed	0.172	0.201	0.185
	CCL	Primary	<b>0.708</b>	<b>0.687</b>	<b>0.697</b>
		Auxiliary	<b>0.729</b>	<b>0.642</b>	<b>0.683</b>
		Attributed	<b>0.646</b>	<b>0.627</b>	<b>0.636</b>
ICT	GPT-3.5-turbo	Primary	0.682	0.625	0.652
		Auxiliary	0.321	0.290	0.305
		Attributed	0.347	0.309	0.327
	Simplified CCL	Primary	0.469	0.567	0.514
		Auxiliary	0.287	0.292	0.289
		Attributed	0.233	0.235	0.234
	CCL	Primary	<b>0.917</b>	<b>0.917</b>	<b>0.917</b>
		Auxiliary	<b>0.855</b>	<b>0.824</b>	<b>0.839</b>
		Attributed	<b>0.810</b>	<b>0.805</b>	<b>0.807</b>

### 4.3 Results

We evaluate CCL and baselines on two datasets through ten-fold cross-validation and the overall results are shown in Table 2. CCL performs best in all evaluation metrics and even outperforms GPT-3.5-turbo by 33.9% and 48.0% in F1-score on two datasets.

We also find that all extraction is better performed on ICT dataset. One important reason is that open dataset is built by means of some fixed mapping rules without considering semantic correctness, which not only has bad effects on model training but may also cause wrong judgment on outputs due to incorrect annotations. In addition, entities in open data is more complicated while most entities in the ICT dataset are merely names of devices, which reduces difficulty in relation extraction.

It’s noteworthy that GPT’s suboptimal performance may stem from the inherent traits of generative models. We observe GPT often produces semantically valid relations that fall outside predefined ranges. While this is minor for general reading, it poses challenges in data structuring and normalization, requiring additional alignment efforts. Consequently, CCL proves more adept for structuring text data in industrial contexts.

### 4.4 Ablation Study

To assess the impact of individual modules on extraction performance, we performed ablation studies on the contrastive learning module, combination module, and parameter-sharing mechanism. Table 3 depicts the performance variations in attributed

triple extraction on the ICT dataset upon sequential removal of each component. Results indicate that all these modules positively influence attributed triple extraction. Model<sub>4</sub> is actually the simplified CCL.

**Table 3.** Results of ablation study.

Name	Detail	Prec.	Rec.	F1
Model <sub>1</sub>	CCL	<b>0.810</b>	<b>0.805</b>	<b>0.807</b>
Model <sub>2</sub>	Model <sub>1</sub> -Contrastive Learning	0.652	0.583	0.616
Model <sub>3</sub>	Model <sub>2</sub> -Combination	0.429	0.402	0.415
Model <sub>4</sub>	Model <sub>3</sub> -Parameter Sharing	0.233	0.235	0.234

**Effect of Contrastive Learning.** The results in the first two rows of Table 3 indicate that the contrastive learning module does enhance the model’s capability of representation with limited training data, leading to an improvement of 19.1% in F1-score when extracting attributed triples. Extra experiments demonstrate that this module also enhances CCL’s performance in extracting primary and auxiliary relational information by 7.8% and 13.1% in F1-score respectively, which is shown in Table 4.

**Table 4.** Detailed results of ablation on contrastive learning.

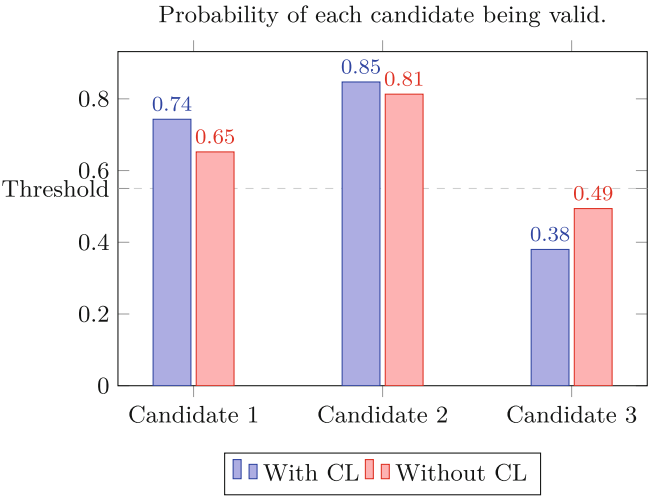
Model	Composition	Prec.	Rec.	F1
CCL	Primary triple	<b>0.917</b>	<b>0.917</b>	<b>0.917</b>
	Auxiliary pair	0.855	0.824	0.839
CCL-Contrastive Learning	Primary triple	<b>0.832</b>	<b>0.814</b>	<b>0.823</b>
	Auxiliary pair	0.705	0.679	0.692

**Effect of Combination.** Results in Table 3 show that replacing combination module by simple rules like nearest-neighbor criterion brings a decline of 20.1% in F1-score, which means that the combination module is effective. In fact, we find that combination can be replaced by more complicated rules that contain common combination patterns. But it takes too much time to manually summarize such rules and rules may fail when encountering unseen combination patterns. Hence, combination module is efficient in industrial scenarios, especially when the semantics are complicated.

**Table 5.** Results of entity recognition w/o parameter sharing.

Model	Prec.	Rec.	F1
CCL	<b>0.931</b>	<b>0.923</b>	<b>0.927</b>
CCL-Parameter Sharing	0.543	0.521	0.532

**Effect of Parameter Sharing.** Parameter sharing mechanism is designed to increase the training of entity recognition module when in lack of training data. According to ablation results, parameter sharing mechanism improves F1-score by 18.1% on ICT dataset. To further verify its effect on entity recognition, we compare the performance of entity recognition of our model on ICT dataset when with and without parameter sharing. The result in Table 5 proves the effectiveness of this mechanism.



**Fig. 3.** Effect of contrastive learning on combination. CL is short for contrastive learning in legend. We use candidates shown in combination module in Fig. 2.

4.5 Case Study

In order to illustrate the advantages of combination under contrastive learning more intuitively, we conduct a case study by comparing results of combination module w/o contrastive learning. Figure 3 shows the visualized effect of contrastive learning on the probability of each candidate combination being valid. It can be observed that valid combinations are more likely to be accepted and invalid combinations are more likely to be denied with contrastive learning. This reflects that our model learns a more reasonable representation space with same training data, which is of great use when the semantics are puzzling.

## 5 Conclusion

In this paper, we presented a novel representation of complex knowledge, termed **attributed triple**, and develop a corresponding extraction model. The attributed triple is designed to intricately detail both the collective and specific relations among multiple entities. This extraction model identifies relations and then leverages these identified relations to extract relevant entities, thereby generating the primary triple and auxiliary pairs. It effectively overcomes the limitations imposed by entity-type pre-definition. Additionally, a linear combination discriminator is utilized to ascertain the semantic feasibility of the combinations formed by these primary triples and auxiliary pairs. To robustly evaluate the model's proficiency in extracting attributed triple data, we constructed a high-quality dataset derived from ICT corpus. Our model significantly outperforms established baselines across a range of metrics.<sup>1</sup>

## References

1. Al-Moslmi, T., Ocaña, M., Opdahl, A., et al.: Named entity extraction for knowledge graphs: a literature overview. *IEEE Access* **8**, 32862–32881 (2020)
2. Ma, X.: Knowledge graph construction and application in geosciences: a review. *Comput. Geosci.* **161**, 105082 (2022)
3. Zheng, H., Wen, R., Chen, X., et al.: PRGC: Potential relation and global correspondence based joint relational triple extraction. In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (Volume: Long Papers)* **1**, 6225–6235 (2021)
4. Peng, N., Poon, H., Quirk, C., et al.: Cross-sentence n-ary relation extraction with graph LSTMs. *Trans. Assoc. Comput. Linguist.* **5**, 101–115 (2017)
5. Song, L., Zhang, Y., Wang, Z., et al.: N-ary relation extraction using graph-state LSTM. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2226–2235 (2018)
6. Guo, Z., Zhang, Y., Lu, W.: Attention guided graph convolutional networks for relation extraction. In: *Proceedings of the 57th Conference of the Association for Computational Linguistics, Vol.1, Long Papers*, pp. 241–251 (2019)
7. Zeng, D., Zhao, C., Jiang, C., et al.: Document-level relation extraction with context guided mention integration and inter-pair reasoning. *IEEE ACM Trans. Audio Speech Lang. Process* **31**, 3659–3666 (2023)
8. Shang, Y., Huang, H., Sun, X., et al.: Relational triple extraction: one step is enough. In: Luc De Raedt, editor, *In: Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, pp. 4360–4366 (2022)
9. Shang, Y., Huang, H., Mao, X.: Onerel: joint entity and relation extraction with one module in one step. In: *Thirty-Sixth AAAI Conference on Artificial Intelligence*, pp. 11285–11293 (2022)
10. Wei, Z., Su, J., Wang, Y., et al.: A novel cascade binary tagging framework for relational triple extraction. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1476–1488 (2020)

<sup>1</sup> This work is partially supported by National Nature Science Foundation of China under No. U21A20488. We thank the Big Data Computing Center of Southeast University for providing the facility support on the numerical calculations in this paper.

11. Panchendrarajan, R., Amaresan, A.: Bidirectional LSTM-CRF for named entity recognition. In: Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation (2018)
12. Ye, H., Zhang, N., Deng, S., et al.: Contrastive triple extraction with generative transformer. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, 14257–14265 (2021)
13. Chen, T., Kornblith, S., Norouzi, M. et al.: A simple framework for contrastive learning of visual representations. In: Proceedings of the 37th International Conference on Machine Learning, vol. 119 of Proceedings of Machine Learning Research, pp. 1597–1607 (2020)