# Query-based Instance Discrimination Network for Relational Triple Extraction

**Zeqi Tan**[1], **Yongliang Shen**[1], **Xuming Hu**[2], **Wenqi Zhang**[1], **Xiaoxia Cheng**[1], **Weiming Lu**[1]*, **Yueting Zhuang**[1]
[1]Zhejiang University, [2]Tsinghua University
[1]{zqtan,syl,luwm,yzhuang}@zju.edu.cn
[2]{hxm19}@mails.tsinghua.edu.cn

## Abstract

Joint entity and relation extraction has been a core task in the field of information extraction. Recent approaches usually consider the extraction of relational triples from a stereoscopic perspective, either learning a relation-specific tagger or separate classifiers for each relation type. However, they still suffer from error propagation, relation redundancy and lack of high-level connections between triples. To address these issues, we propose a novel query-based approach to construct instance-level representations for relational triples. By metric-based comparison between query embeddings and token embeddings, we can extract all types of triples in one step, thus eliminating the error propagation problem. In addition, we learn the instance-level representation of relational triples via contrastive learning. In this way, relational triples can not only enclose rich class-level semantics but also access to high-order global connections. Experimental results show that our proposed method achieves the state of the art on five widely used benchmarks.

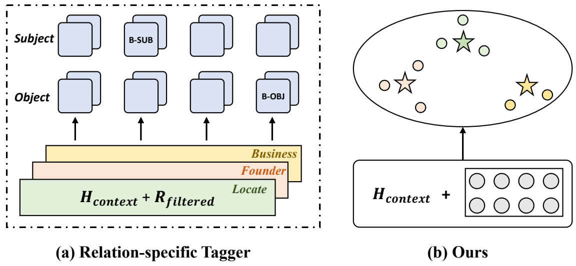


Figure 1: (a) $H_{context}$ represents sentence representation, $R_{potential}$ represents potential relation type embeddings. PRGC first filters out unlikely relations in a sentence, and then learns the relation-specific tagger for potential relations. (b) The gray circle below represents the initial state of query embeddings which are type-agnostic, and the star above represents relation type embeddings. We can break the limitation of type-independence and learn the high-order connections between different types via contrastive learning.

## 1 Introduction

Extracting structured information from open domain texts is a long-standing study topic in NLP. Joint entity and relation extraction aims to mine high-quality relational triples (*subject, relation, object*) from unstructured texts. For example, in the sentence "900 people cross the border between Malaysia and Singapore", the subject entity *900 people* and the object entity *border* have a physical location relation PHYS.

The current entity and relation extraction methods can be divided into two categories: pipeline methods (Chan and Roth, 2011; Lin et al., 2016) and joint methods (Miwa and Bansal, 2016; Katiyar and Cardie, 2017). Despite the flexibility of pipeline methods, they face the error propagation and the lack of interaction problems (Katiyar and Cardie, 2017; Wang and Lu, 2020). Therefore, many strategies have been proposed to unify the two tasks. Zheng et al. (2017) extends tagging schema to tag both entities and relations. However, it cannot handle the triple overlapping problem.

To tackle this problem, the generation-based methods (Zeng et al., 2018, 2020; Nayak and Ng, 2020; Ye et al., 2021), the table-filling methods (Wang et al., 2020b; Wang and Lu, 2020; Wang et al., 2021) and the cascading methods (Luan et al., 2019; Yuan et al., 2020; Wei et al., 2020; Zheng et al., 2021) have been investigated.

Recently, an increasing number of studies have come to consider the relational triple extraction task from a stereoscopic perspective. These approaches treat relation type as an important element, and either learn a distinct classifier (Liu et al., 2020; Wang et al., 2020b) for each relation type or construct a relation-specific sequence tagger (Yuan et al., 2020; Wei et al., 2020; Zheng et al., 2021). In this way, these methods can better decouple the task-specific features for relations and entities, and

* Corresponding author



*Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7677–7690
December 7-11, 2022 ©2022 Association for Computational Linguistics

yield promising results. However, they still have some weaknesses. **First**, error propagation is a well-known problem. Wei et al. (2020) first identifies the subject and then extracts the corresponding triples. However, there are inevitable errors in the first stage. PRGC (Zheng et al., 2021), as shown in Figure 1 (a), first picks the potential relations in the sentence, and later tags the relation-specific entities. The correct relations are then likely to be filtered out in this process. **Second**, these methods mostly suffer from the relation redundancy problem. Wei et al. (2020); Wang et al. (2020b) assembles triples on all relation types, which generates numerous invalid operations, leading to sparse label and low convergence rate. **Third**, these methods process the triples for each relation type independently, ignoring the high-level global connections between different types of triples. For example, if the triple (*Maryland, country, U.S.*) has been recognized, then *U.S.* is not supposed to appear in any other family relationship, such as *spouse of*.

To tackle the above problems, we propose a novel query-based instance discrimination network (QIDN), in which we leverage instance query to construct instance-level representation for different types of relational triples. As in Figure 1 (b), we first use a set of type-agnostic query embeddings to obtain useful contextual information from a pre-trained language model. Afterwards, by the metric-based comparison between query embeddings and token embeddings, we can extract triples of all types in one step. In this manner, the error propagation problem is eliminated. Moreover, since we query the most plausible relation type directly with the query vector, the problem of relation redundancy is well mitigated. In addition, we construct instance-level representations for relational triples by contrastive learning. Specifically, as shown in Figure 1 (b), we set two training objectives: (1) Intra-class instance pairs should get higher similarity than inter-class ones. (2) Instance representations should be closer to their corresponding relation type representations. We aim to break the limitation of type-independence in previous methods and establish the global connection between triples.

Our main contributions are as follows:

- We tackle the relational triple extraction task from a novel perspective, focusing on constructing instance-level representations for relational triples. Based on a simple similarity measure, we extract all types of triples in one step, thus avoiding cascading errors.
- We learn the instance-level representation of relational triples by contrastive learning. In this way, relational triples are not only able to establish high-order global connections, but also to enclose rich class-level semantics.
- Extensive experiments on five public benchmarks demonstrate that our model achieves state-of-the-art, and significantly outperforms a range of robust baselines.

## 2 Related Work

### 2.1 Mainstream methods

Relational triple extraction can be solved with two different models. Assuming that the entities in a text sequence are obtained by NER model (Zhang et al., 2021; Tan et al., 2021; Shen et al., 2022; Zhang et al., 2022), relation extraction can be considered as a classification task (Zeng et al., 2014; Wang et al., 2019; Hu et al., 2020, 2021). Since these methods require additional entity annotators to carry out a pipeline, they generally face the error propagation problem and the lack of interaction between tasks (Lin et al., 2016; Shen et al., 2021a). Recently, Zhong and Chen (2021) propose a simple pipeline approach reaching state-of-the-art. They re-construct the input text of the relation model with the entity recognition results, delivering the type and location info of entities in this manner.

Beside the pipeline methods, the joint methods can be divided into the generation-based methods (Zeng et al., 2018, 2020; Nayak and Ng, 2020; Ye et al., 2021), the table-filling methods (Gupta et al., 2016; Zhang et al., 2017; Wang et al., 2020b, 2021) and the cascading methods (Luan et al., 2019; Yuan et al., 2020; Wei et al., 2020; Zheng et al., 2021). The generation-based methods use a sequence-to-sequence model to generate the relation triples directly from the sentence, while the table-filling methods treat on/off-diagonal entries as labels for entities and relationships, respectively. Differently, Luan et al. (2019) treats entity and relation extraction as a cascading two-level span classification task. Later, some work introduce a relation-specific perspective to treat triple extraction as a relation-first (Yuan et al., 2020; Zheng et al., 2021) or relation-middle (Wei et al., 2020) cascading process. Compared to them, our method can form triples in one step and avoid the cascading errors.

### 2.2 Query-based methods

To exploit the well developed machine reading comprehension models, a number of works (Li et al., 2019, 2020; Zhao et al., 2020; Du and Cardie, 2020) extend the MRC paradigm to information extraction tasks. Li et al. (2019) recast the entity and relation extraction task as a multi-turn question answering problem. Their queries are constructed based on pre-defined templates and the answers are the entity spans in the sentence. Zhao et al. (2020) explores to cover more relational semantics with more handcrafted query templates based on subcategories of relations.

Different from the above methods where queries are manually constructed, in the field of object detection in computer vision, there is a range of work (Carion et al., 2020; Zhu et al., 2020; Sun et al., 2020) that uses trainable query to learn task-specific features automatically. DETR (Carion et al., 2020) successfully integrates Transformer (Vaswani et al., 2017), originally designed for text, by using a set of learnable queries. These queries are type-agnostic and have the potential to build global dependencies between objects at diverse scales. Inspired by this, we employ such query embeddings to construct instance-level representations for relational triples and establish global connections between different types of triples.

### 2.3 Contrastive relation learning

MTB (Soares et al., 2019) extracts relation-aware semantics from text by comparing sentences with the same entity pairs. Extended MTB, CP (Peng et al., 2020) samples relational triples with better diversity and increases the coverage of entity types and different contexts. Different from the sentence-level relational contrastive learning (Soares et al., 2019; Peng et al., 2020; Liu et al., 2022), ERICA (Qin et al., 2021) proposes a relation discrimination task to distinguish whether two relation types are semantically similar or not, better considering the interaction between multiple relations. Compared to them, we use query embeddings to construct instance-level triple representations, taking into account both entity features and relation features, which may motivate new pre-training tasks.

## 3 Method

### 3.1 Task Formulation

The entity and relation extraction task aims to extract a set of entities and a set of relations from the text. Formally, given an input sentence $X = x_1, x_2, \dots, x_n$ ($x_i$ is the i-th token, $n$ is the sentence length), an entity in the entity set $\mathcal{E}$ is denoted as $(x_i, x_j, t_e)$, where $x_i$ and $x_j$ are the left and right token of the entity with a pre-defined entity type $t_e$ in $\mathcal{Y}_e$. For the relation set $\mathcal{R}$, a relation is denoted as $(e_1, e_2, t_r)$, where $e_1, e_2 \in \mathcal{E}$ are the subject and object entities and $t_r$ is a pre-defined relation type in $\mathcal{Y}_r$. Besides, we add an additional label $\varnothing$ to $\mathcal{Y}_e$ and $\mathcal{Y}_r$ to indicate that no entity or no relation is recognized.

### 3.2 Sentence Encoder

As shown in Figure 2, given the input sentence $X$, we first use the pre-trained BERT model (Devlin et al., 2019) to obtain a contextual representation for each token. Afterwards, to better consider the order information of the tokens, we feed the token representations into BiLSTM (Zhang et al., 2015) to get the final sentence representation $H \in \mathbb{R}^{n \times d}$, where $n$ is sentence length and $d$ is hidden size.

### 3.3 Triple Prediction

In our method, we employ a set of learnable instance queries same as DETR (Carion et al., 2020), which are denoted as $Q = \mathbb{R}^{M \times d}$. Each query (denoted as a vector of size $d$) is responsible for extracting one relational triple. These queries are randomly initialized and the number of the queries $M$ is pre-specified. Different from DETR, in order to distinguish entity and relation task-specific features, we project the queries $Q$ into entity and relation branches by FFN before entering the decoder. After decoding, we use several prediction heads to map these query vectors to different representation spaces for metric-based comparison with the token embeddings. In this way, all types of triples can be recognized in one step.

**Transformer-based Decoder** The decoder is composed of a stack of $L$ transformer (Vaswani et al., 2017) layers, and the decoding process mainly involves the multi-head attention mechanism. For simplicity, we denote the attention as:

$$\text{Attention}\,(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{1}$$

where $Q$, $K$, $V$ are the query, key and value matrix respectively, and the $1/\sqrt{d_k}$ is the scaling factor. In the cross-attention mechanism module, we do not directly take $H$ from the sentence encoder as

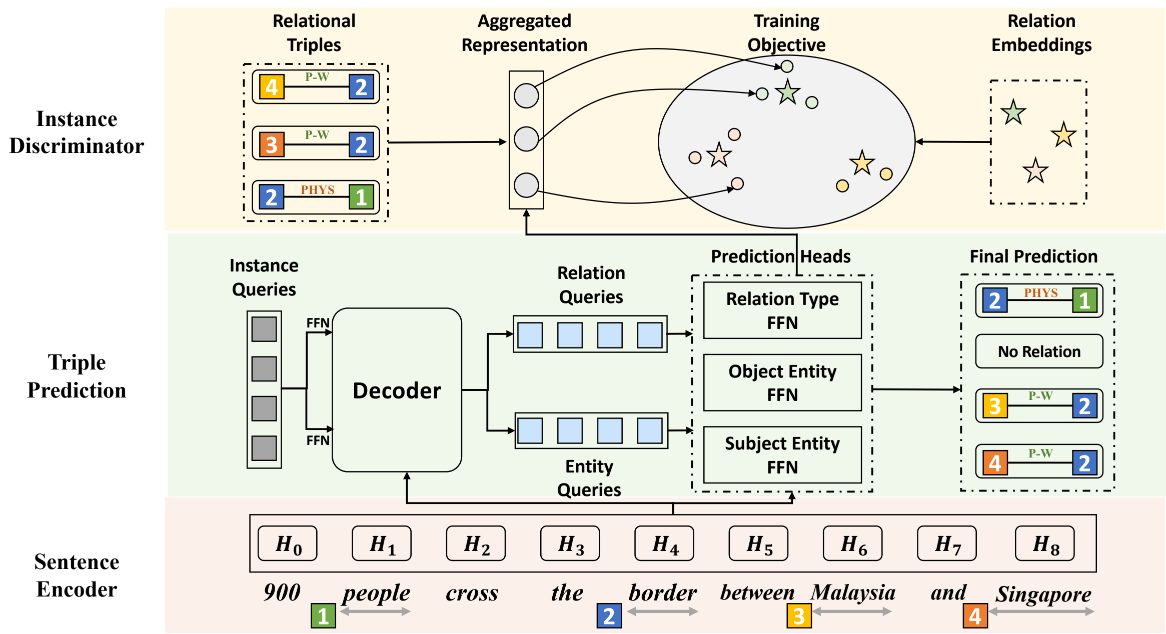


Figure 2: The overall architecture of our method. First we utilize the sentence encoder to get the context representation. Then, we employ a transformer-based decoder to transform a set of instance queries. Afterwards, we use several prediction heads to map the query vectors to different representation spaces for metric-based comparison with the token embeddings, so that all triples can be recognized in one step. Furthermore, we aggregate the representation of these prediction heads to construct instance-level representations for triples. By setting two contrastive learning training objectives, we can capture global connections between triples and rich class-level semantic information.

key and value, instead, we construct span-level representations to obtain hierarchical semantic information. Let $S = s_1, s_2, \ldots, s_{n_s}$ be all possible spans in sentence $X$ of up to limited length ($n_s$ is the number of spans). Give a span $s_i \in S$, the span representation $H_{\mathrm{i}}^{span}$ is defined as:

$$H_{\mathrm{i}}^{span} = \left[H_{\mathrm{start(i)}}; H_{\mathrm{end(i)}}; \phi\left(s_{\mathrm{i}}\right)\right], \tag{2}$$

where $[;]$ denotes concatenation operation, $H_{\mathrm{start(i)}}$ and $H_{\mathrm{end(i)}}$ are the representations of the boundary tokens. $\phi\left(s_{\mathrm{i}}\right)$ denotes the span-length feature (Zhong and Chen, 2021). Then, the span-level representations is denoted as $H^{span} \in \mathbb{R}^{n_s \times \mathrm{d}}$. Through the $L$ decoding layers, instance queries $Q$ are decoded into $Q_r$ and $Q_e$ as:

$$[Q_r; Q_e] = \mathrm{Decoder}\left([QW_r; QW_e], H^{span}\right), \tag{3}$$

where $W_r, W_e \in \mathbb{R}^{d \times d}$ are trainable parameters, $Q_r, Q_e \in \mathbb{R}^{M \times d}$ denote relation and entity queries. For a more intuitive understanding, we illustrate the architecture of the decoder in Appendix B.

**Relation Head** For relation queries $Q_r$, we feed them into a single layer of FFN to predict the categories of their corresponding triples. Formally, we define the probability of the $i$-th query belonging to type $c$ as:

$$P_{ic}^{t} = \frac{\exp(Q_r^i W_t^c + b_t^c)}{\sum_{c'}^{|\mathcal{Y}_r|} \exp(Q_r^i W_t^{c'} + b_t^{c'})}, \tag{4}$$

where $W_t \in \mathbb{R}^{|\mathcal{Y}_r| \times d}$ and $b_t \in \mathbb{R}^{|\mathcal{Y}_r|}$ are trainable parameters.

**Entity Head** To predict the boundary tokens of triples, we first perform linear projections for the entity queries $Q_e$ and the token representations $H$ by a single layer of FFN as:

$$E_\delta = Q_e W_\delta, H_s = HW_s, \tag{5}$$

where $\delta \in \mathcal{C} = \{l_{sub}, r_{sub}, l_{obj}, r_{obj}\}$ denotes the left or right boundaries of subject or object entities and $W_\delta, W_s \in \mathbb{R}^{d \times d}$ are projection parameters. To measure the similarity between them, we adopt cosine similarity function $S(\cdot)$ as:

$$S\left(\mathbf{v}_i, \mathbf{v}_j\right) = \frac{\mathbf{v}_i}{\|\mathbf{v}_i\|} \cdot \frac{\mathbf{v}_j}{\|\mathbf{v}_j\|}. \tag{6}$$

Then, we can calculate the probability that the $j$-th token is the boundary token of the $i$-th entity queries as:

$$P_{ij}^{\delta} = \frac{\exp S\left(E_{\delta}^{i}, H_{s}^{j}\right)}{\sum_{j'}^{n} \exp S\left(E_{\delta}^{i}, H_{s}^{j'}\right)}, \tag{7}$$

where $n$ is the number of tokens in the sentence. Finally, based on the probability $P^t$ and $P^\delta$ ($\delta \in \mathcal{C}$), we can predict all types of triples. Besides, for corpus with labeled entity types, we add two additional entity type heads like the relation head.

### 3.4 Instance Discriminator

In this module, we first aggregate several representations from the prediction heads to form the initial representation of triples, and then we set up two contrastive learning objectives to build global connections between triple instances and learn rich class-level semantic information.

**Aggregation** To get the initial representation of triples, we first use FFN to project the relation queries $Q_r$, aligned with $E_\delta$ from the entity head. Then, we aggregate category representations and boundary representations by a simple summing operation. The triple instance representation $\mathbf{v}$ is:

$$\mathbf{v} = Q_r W + \sum_{\delta \in \mathcal{C}} E_\delta. \tag{8}$$

**Training Objective** With the triple instance representation $\mathbf{v}$ and the similarity function defined in equation 6, we discuss how to set our training objectives. Denote $\mathcal{R} = \left\{\mathbf{r}_1, \cdots, \mathbf{r}_{|\mathcal{Y}_r|}\right\}$ as the set of relation type embeddings which are randomly initialized. We expect to achieve two goals with optimization: (1) For instance-instance pairs, intra-class pairs should get higher similarity than inter-class ones. (2) For instance-type pairs, instance representations should be closer to their corresponding relation type representations. To reach these two goals, we define the objective function based on InfoNCE (van den Oord et al., 2018), which is widely applied in contrastive learning.

For the instance-instance objective, we minimize the following loss:

$$\mathcal{L}_{\text{ins}} = -\sum_{c}\sum_{i,j} \log \frac{\exp S\left(\mathbf{v}_i^c, \mathbf{v}_j^c\right)}{\sum_{c',j'} \exp S\left(\mathbf{v}_i^c, \mathbf{v}_{j'}^{c'}\right)}, \tag{9}$$

where $\left(\mathbf{v}_i^c, \mathbf{v}_j^c\right)$ denotes the instance pair of the same type $c$. Similarly, the instance-type loss is defined as:

$$\mathcal{L}_{\text{cls}} = -\sum_{i,c} \log \frac{\exp S\left(\mathbf{v}_i^c, \mathbf{r}_c\right)}{\sum_{c'} \exp S\left(\mathbf{v}_i^c, \mathbf{r}_{c'}\right)}, \tag{10}$$

where $\mathbf{v}_i^c$ is the instance of type $c$ and $\mathbf{r}_c \in \mathcal{R}$ is the relation embedding corresponding to type $c$. We aim to break the limitation of type-independence and establish the global connection between different types of triples.

| **Dataset** | **#Sentences** | | | **Relations** |
|---|---|---|---|---|
| | **Train** | **Dev** | **Test** | |
| NYT | 56,195 | 4,999 | 5,000 | 24 |
| WebNLG | 5,019 | 500 | 703 | 170 |
| NYT* | 56,195 | 5,000 | 5,000 | 24 |
| ACE05 | 10,051 | 2,424 | 2,050 | 6 |
| SciERC | 1,861 | 275 | 551 | 7 |

Table 1: Dataset statistics.

### 3.5 Training and Inference

**Training** In training, we define the triple loss $\mathcal{L}_{tri}$ according to the type probability $P^t$ and the boundary probability $P^\delta$ as:

$$\mathcal{L}_{\text{tri}} = -\sum_{i=1}^{M}\left(\log P_{\sigma(i)}^{t} + \sum_{\delta \in \mathcal{C}} \log P_{\sigma(i)}^{\delta}\right), \tag{11}$$

where $M$ is the number of instance queries, $\sigma$ is the optimum matching calculated as same as Carion et al. (2020). The final loss function $\mathcal{L}$ is computed as $\mathcal{L} = \mathcal{L}_{tri} + \mathcal{L}_{ins} + \mathcal{L}_{cls}$.

**Inference** During inference, the triple predicted by the $i$-th instance query is $\mathcal{Y}_i = \left(\mathcal{Y}_i^t, \mathcal{Y}_i^\delta\right)$, $\delta \in \mathcal{C}$. $\mathcal{Y}_i^t = \arg\max_c(P_{ic}^t)$ is the relation type. $\mathcal{Y}_i^\delta = \arg\max_k(P_{ik}^\delta)$ are the left and right boundary tokens of subject-object entities, and triples with the predicted type of $\varnothing$ will be filtered out.

## 4 Experiments

### 4.1 Experimental Setup

**Datasets** We conduct our experiments on four widely used datasets: NYT (Riedel et al., 2010), WebNLG (Zeng et al., 2018), ACE05 (Walker et al., 2006), and SciERC (Luan et al., 2018). NYT (Riedel et al., 2010) is sampled from New York Times news articles and annotated by distant supervision. NYT* is another version of it which annotates the whole span of entities. WebNLG is originally created for Natural Language Generation (NLG) and is adopted by (Zeng et al., 2018)

| Model | NYT | | | WebNLG | | | NYT* | | |
|---|---|---|---|---|---|---|---|---|---|
| | Prec. | Rec. | F1 | Prec. | Rec. | F1 | Prec. | Rec. | F1 |
| GraphRel (Fu et al., 2019) | 63.9 | 60.0 | 61.9 | 44.7 | 41.1 | 42.9 | - | - | - |
| RSAN (Yuan et al., 2020) | - | - | - | - | - | - | 85.7 | 83.6 | 84.6 |
| MHSA (Liu et al., 2020) | 88.1 | 78.5 | 83.0 | 89.5 | 86.0 | 87.7 | - | - | - |
| CasRel (Wei et al., 2020) | 89.7 | 89.5 | 89.6 | 93.4 | 90.1 | 91.8 | - | - | - |
| TPLinker (Wang et al., 2020b) | 91.3 | 92.5 | 91.9 | 91.8 | 92.0 | 91.9 | 91.4 | 92.6 | 92.0 |
| SPN (Sui et al., 2020) | 93.3 | 91.7 | 92.5 | 93.1 | 93.6 | 93.4 | 92.5 | 92.2 | 92.3 |
| CGT (Ye et al., 2021) | **94.7** | 84.2 | 89.1 | 92.9 | 75.6 | 83.4 | - | - | - |
| CasDE (Ma et al., 2021) | 90.2 | 90.9 | 90.5 | 90.3 | 91.5 | 90.9 | 89.9 | 91.4 | 90.6 |
| RIFRE (Zhao et al., 2021) | 93.6 | 90.5 | 92.0 | 93.3 | 92.0 | 92.6 | - | - | - |
| PRGC (Zheng et al., 2021) | 93.3 | 91.9 | 92.6 | 94.0 | 92.1 | 93.0 | **93.5** | 91.9 | 92.7 |
| QIDN | 93.4 | **92.6** | **93.0** | **94.1** | **93.7** | **93.9** | 93.3 | **92.5** | **92.9** |

Table 2: Precision(%), Recall (%) and F1-score (%) of our method and baselines. Note that the sentence encoders adopted in GraphRel, RSAN and MHSA are LSTM networks, while other baselines employ the BERT model.

as a relation extraction dataset. ACE05 corpora are collected from a wide range of domains, such as newswire and online forums. SciERC includes annotations for scientific entities, their relations, and coreference clusters for 500 scientific abstracts. Table 1 shows the detailed dataset statistics.

**Evaluation Metrics** Following previous works (Wei et al., 2020; Zheng et al., 2021), we adopt micro Precision (Prec.), Recall (Rec.) and F1-score (F1) on NYT/WebNLG under partial match, on NYT* under exact match. For ACE05 and SciERC, same as (Zhong and Chen, 2021; Yan et al., 2021), we adopt the micro F1-score as the evaluation metric for both NER and RE. We use the strict evaluation criterion that a relation is correct only if the relation type is correct and the type as well as boundaries of its corresponding subject-object entities are correct.

**Implementation Details** For fair comparison with prior work, we use *bert-base-cased* (Devlin et al., 2019) for NYT/WebNLG, *albert-xxlarge-v1* (Lan et al., 2020) for ACE05, and *scibert-scivocab-uncased* (Beltagy et al., 2019) for SciERC in the sentence encoder. The instance queries and relation embeddings are randomly initialized with the normal distribution $\mathcal{N}(0.0, 0.02)$ and we design a comparison experiment in Appendix A to determine the query number $M$ as 15. The number of the BiLSTM layer is 3 and the number of our decoder layer $L$ is set to 5. The span enumeration length is set to 8. We adopt a batch size of 8 on NYT and ACE05, and 4 on WebNLG and SciERC in training. We use the AdamW (Loshchilov and Hutter, 2017) optimizer with a linear warmup-decay learning rate schedule. The peak learning rate is set to 1e-5 for the pre-trained model and 3e-5 for the other parameters. We train each model for 100 epochs with a dropout of 0.1 (Srivastava et al., 2014).

| Model | NER | RE |
|---|---|---|
| **ACE05** | | |
| Structured Perceptron (Li and Ji, 2014) | 80.8 | 49.5 |
| SPTree (Miwa and Bansal, 2016) | 83.4 | 55.6 |
| Multi-turn QA (Li et al., 2019) † | 84.8 | 60.2 |
| Table-Sequence (Wang and Lu, 2020) ‡ | 89.5 | 64.3 |
| Trigger-Sense (Shen et al., 2021b) † | 87.6 | 62.8 |
| PURE (Zhong and Chen, 2021) ‡ | 89.7 | 65.6 |
| UniRE (Wang et al., 2021) ‡ | 90.2 | 66.0 |
| PFN (Yan et al., 2021) ‡ | 89.0 | 66.8 |
| QIDN ‡ | **90.5** | **68.2** |
| **SciERC** | | |
| SPE (Wang et al., 2020a) § | 68.0 | 34.6 |
| PURE (Zhong and Chen, 2021) § | 66.6 | 35.6 |
| UniRE (Wang et al., 2021) § | 68.4 | 36.9 |
| PFN (Yan et al., 2021) § | 66.8 | 38.4 |
| QIDN § | **69.8** | **39.5** |

Table 3: The overall performances of our method on ACE05 and SciERC. †, ‡ and § denotes the use of BERT, ALBERT and SciBERT(Devlin et al., 2019; Lan et al., 2020; Beltagy et al., 2019) pre-trained language models.

## 4.2 Overall Performance

Table 2 shows the overall performance of our proposed method (QIDN) as well as the baseline models on the NYT and WebNLG datasets. Overall, our method outperforms all the baseline model consistently, and achieves the new state-of-the-art. Compared with the robust baseline models CasRel (Wei et al., 2020) and TPLinker (Wang et al., 2020b), our model makes a significant improvement of +2.1% and +2.0% in absolute F1-measure

| Model | NYT | | | | | | | | | WebNLG | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Normal | EPO | SEO | SOO | N=1 | N=2 | N=3 | N=4 | N≥5 | Normal | EPO | SEO | SOO | N=1 | N=2 | N=3 | N=4 | N≥5 |
| CasRel | 87.3 | 92.0 | 91.4 | 77.0§ | 88.2 | 90.3 | 91.9 | 94.2 | 83.7 | 89.4 | 94.7 | 92.2 | 90.4§ | 89.3 | 90.8 | 94.2 | 92.4 | 90.9 |
| TPLinker | 90.1 | 94.0 | 93.4 | 90.1§ | 90.0 | 92.8 | 93.1 | 96.1 | 90.0 | 87.9 | 95.3 | 92.5 | 86.0§ | 88.0 | 90.1 | 94.6 | 93.3 | 91.6 |
| SPN | 90.8 | 94.1 | 94.0 | - | 90.9 | 93.4 | **94.2** | 95.5 | 90.6 | - | - | - | - | 89.5 | 91.3 | **96.4** | 94.7 | 93.8 |
| PRGC | **91.0** | 94.5 | 94.0 | 81.8 | **91.1** | 93.0 | 93.5 | 95.5 | 93.0 | 90.4 | **95.9** | 93.6 | 94.6 | 89.9 | 91.6 | 95.0 | 94.8 | 92.8 |
| QIDN | **91.2** | **94.9** | **94.8** | **90.7** | 90.6 | **93.6** | 94.1 | **95.8** | **94.3** | **91.5** | 95.4 | **94.8** | **94.9** | **91.2** | **92.8** | 96.1 | **95.4** | **94.2** |

Table 4: F1-measure (%) on sentences with different overlapping patterns and different triple numbers. § marks the results reported by (Zheng et al., 2021).

on WebNLG. Compared to the best cascade method PRGC (Zheng et al., 2021), our method achieves consistent improvement in terms of all three evaluation metrics on WebNLG. Similarly, Table 3 indicates that our method reaches state-of-the-art on both the NER and RE tasks for ACE05 and SciERC. In comparison to the best joint method (Yan et al., 2021), we achieve superior performance in F1-measure on ACE05 and SciERC (+1.5% and +3.0% for NER, +1.4% and +1.1% for RE). The experimental results demonstrate the effectiveness of our proposed method on the joint entity and relation extraction task. We believe the main improvement comes from the fact that our approach avoids cascading errors and the relation redundancy problem. The cascading methods (Wei et al., 2020; Zheng et al., 2021) inevitably suffer from cascading errors. And learning separate classifiers for each relation type (Wang et al., 2020b, 2021) causes the problem of relation redundancy. Our approach solves these problems simultaneously and thus achieves promising results.

### 4.3 Analysis on Complex Scenarios

Following Wei et al. (2020); Wang et al. (2020b); Zheng et al. (2021), we evaluate our model on different triple overlapping patterns and different triple numbers. The triple overlapping problem refers to triples sharing the same single entity (SEO, i.e. SingleEntityOverlap) or entity pair (EPO, i.e. EntityPairOverlap). For example, In “Alice and Joe were born in the US”, triples (*Alice, birthplace, USA*) and (*Joe, birthplace, USA*) share the single entity *USA*, while triples (*Alice, birthplace, USA*) and (*Alice, residence, USA*) share the whole pair. The detailed statistic is described in Appendix.

As shown in Table 4, we achieve the best performance on all three overlapping patterns on WebNLG and NYT. In the normal set, we achieve +1.1% improvement in F1-measure on WebNLG, while on NYT the improvement is very slight. We argue that this is because NYT is generated with distant supervision, and annotations are often incomplete, especially for the normal pattern. In addition, as in Table 4, our model performs well for both datasets with different number of triples, especially for the sentences with more than 5 triples (+1.3% on NYT and +1.4% on WebNLG). Compared to previous methods, our approach can break the limitation of type-independence and establish the global connection between different triples, and thus be more robust than baselines when dealing with these challenging complex scenarios.

| Model | NYT | | | WebNLG | | |
|---|---|---|---|---|---|---|
| | P | R | F | P | R | F |
| Default | **93.4** | **92.6** | **93.0** | **94.1** | **93.7** | **93.9** |
| w/o $H_{span}$ | 92.9 | 92.6 | 92.8 | 93.5 | 93.0 | 93.3 |
| w/o $Q_e, Q_r$ | 93.2 | 92.2 | 92.7 | 93.4 | 92.8 | 93.1 |
| w/o $\mathcal{L}_{\text{ins}}$ | 92.0 | 92.5 | 92.2 | 93.0 | 92.8 | 92.9 |
| w/o $\mathcal{L}_{\text{cls}}$ | 92.7 | 92.2 | 92.4 | 93.7 | 92.5 | 93.1 |
| w/o $\mathcal{L}_{\text{ins}}, \mathcal{L}_{\text{cls}}$ | 91.1 | 92.6 | 91.8 | 93.3 | 91.8 | 92.5 |

Table 5: Ablation studies with five different settings of our model on NYT and WebNLG.

### 4.4 Ablation Study

In this section, we take a closer look at the modules in our model that contribute to performance with five settings. (1) **w/o** $H_{span}$: replace the span-level representation with the token-level representation, (2) **w/o** $Q_e, Q_r$: remove entity and relation query branches and use only instance queries, (3) **w/o** $\mathcal{L}_{\textbf{ins}}$: remove the training objective between instance pairs, (4) **w/o** $\mathcal{L}_{\textbf{cls}}$: remove the training objective between instances and relation embeddings, (5) **w/o** $\mathcal{L}_{\textbf{ins}}, \mathcal{L}_{\textbf{cls}}$: remove both contrastive learning training objectives.

From Table 5 we observe that **w/o** $\mathcal{L}_{\textbf{ins}}, \mathcal{L}_{\textbf{cls}}$ leads to the most significant performance decrease in absolute F1-mearsure (-1.2% on NYT and -1.4% on WebNLG) and removing any of them causes visible performance drops, which demonstrates the effectiveness of the two contrastive learning objectives we set. In addition, as show in Table 5, constructing span-level sentence representations brings performance gains (+0.2% on NYT and +0.6% on WebNLG), which indicates that span-level representations contain structured information that is more useful for triple extraction than token-level representations. Similarly, dividing the instance queries into entity and relation branches to facilitate the distinction between their task-specific characteristics, delivering performance improvement. We provide a more detailed analysis for entity and relation branches in Appendix B.

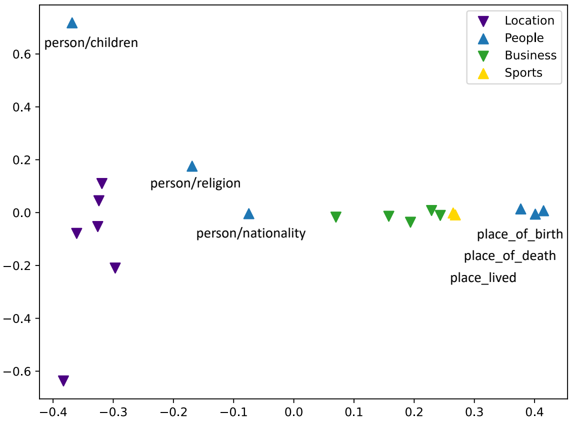


Figure 3: Visualization of relations on NYT dataset. For simplicity, we only label the fine-grained categories under "People".

## 5 Topology of Relations

To illustrate that our approach establishes connections between different types of relations, we visualize the relation representation learned on NYT. We first perform L2 normalization on the relation embeddings and then use PCA (Abdi and Williams, 2010) to reduce their dimension. We filter out 5 long-tail relations that appear less than 40 times in the entire training set and then classify them into 4 broad categories based on label prefixes in NYT. As shown in Figure 3, the topology between the relations indicates the semantic connections between them. For example, the two types of sport relations almost overlap, while the six location relations (including *country*, *capital*, etc.) are all scattered in the lower left corner. However, we observe an exception on "People", where the 6 subcategories are not spread together spatially. We attribute this to semantic divergence. The three subcategories (*place_lived*, *place_of_birth* and *place_of_death*) semantically express position information, which is distinct from the other subcategories. These all demonstrate that our method can break the limits of type-independence and well consider the high-order connections between different relation types. Besides, we conduct a detailed case study to show the connections between queries in Appendix C.

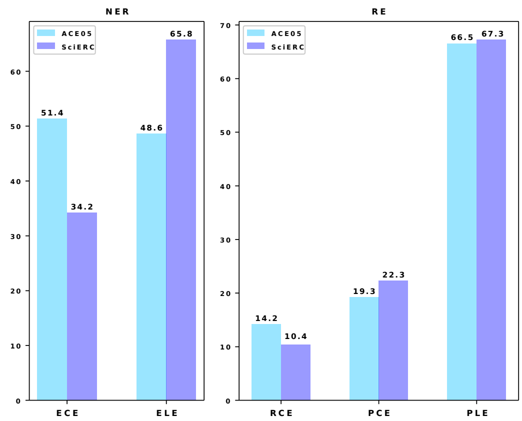


Figure 4: The proportions of different errors in NER and RE on the ACE05 and SciERC test sets.

## 6 Error Analysis

In this section, we analyze the different error proportions on two tasks: entity classification error (ECE) and entity location error (ELE) for NER, relation classification error (RCE), entity-pair classification error (PCE), and entity-pair location error (PLE) for RE. On the NER task, we can observe from Figure 4 that the proportion of classification errors and localization errors for entity queries on ACE05 is comparable. But on SCiERC, the localization errors are more pronounced. We argue that on account of the longer average length of scientific term entities in SciERC, the identification of entity boundaries becomes more difficult.

As shown in Figure 4, identifying the boundary tokens of the subject and object entities is the most difficult on both ACE05 and SciERC. We argue that this is due to the complicated patterns of nesting as well as overlapping in boundary recognition. In addition, the proportion of relation classification errors is minor on both datasets, which indicates the ability of queries in categorization.

## 7 Conclusion

In this paper, we propose a novel query-based instance discrimination network for relational triple extraction. Based on instance queries, we can extract all types of triples in one step, thus avoiding the problem of error accumulation and relation redundancy. Besides, we set two training objectives for contrastive learning to establish high-order connections between triples while learning rich class-level semantic information. Experimental verifications on five datasets demonstrate that the proposed method reaches state-of-the-art.

## Limitations

We discuss here the limitations of the method in this paper. First, this method requires specifying the relation type in advance and cannot handle unseen categories, which is weak for relation extraction in open domains. Second, due to the inherent length limit of BERT, this model cannot deal with excessively long texts. Besides, for the case of a single entity corresponding to multiple mentions in a long document, it can pose a great challenge to the proposed method. Finally, incorporating prior knowledge into query embeddings is a promising direction for optimization.

## Acknowledgments

This work is supported by the National Key Research and Development Project of China (No. 2018AAA0101900), the Key Research and Development Program of Zhejiang Province, China (No. 2021C01013), CKCEST, and MOE Engineering Research Center of Digital Library.

## References

Hervé Abdi and Lynne J Williams. 2010. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. 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)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-end object detection with transformers. *arXiv preprint arXiv:2005.12872*.

Yee Seng Chan and Dan Roth. 2011. Exploiting syntactico-semantic structures for relation extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 551–560, Portland, Oregon, USA. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Xinya Du and Claire Cardie. 2020. Event extraction by answering (almost) natural questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 671–683, Online. Association for Computational Linguistics.

Tsu-Jui Fu, Peng-Hsuan Li, and Wei-Yun Ma. 2019. GraphRel: Modeling text as relational graphs for joint entity and relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1409–1418, Florence, Italy. Association for Computational Linguistics.

Pankaj Gupta, Hinrich Schütze, and Bernt Andrassy. 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*, pages 2537–2547, Osaka, Japan. The COLING 2016 Organizing Committee.

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

Xuming Hu, Chenwei Zhang, Yawen Yang, Xiaohe Li, Li Lin, Lijie Wen, and S Yu Philip. 2021. Gradient imitation reinforcement learning for low resource relation extraction. In *Proc. of EMNLP*, pages 2737–2746.

Arzoo Katiyar and Claire Cardie. 2017. Going out on a limb: Joint extraction of entity mentions and relations without dependency trees. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 917–928, Vancouver, Canada. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Qi Li and Heng Ji. 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)*, pages 402–412, Baltimore, Maryland. Association for Computational Linguistics.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020. A unified MRC framework for named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5849–5859, Online. Association for Computational Linguistics.

Xiaoya Li, Fan Yin, Zijun Sun, Xiayu Li, Arianna Yuan, Duo Chai, Mingxin Zhou, and Jiwei Li. 2019. Entity-relation extraction as multi-turn question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1340–1350, Florence, Italy. Association for Computational Linguistics.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2124–2133, Berlin, Germany. Association for Computational Linguistics.

Jie Liu, Shaowei Chen, Bingquan Wang, Jiaxin Zhang, Na Li, and Tong Xu. 2020. Attention as relation: Learning supervised multi-head self-attention for relation extraction. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3787–3793.

Shuliang Liu, Xuming Hu, Chenwei Zhang, Shu'ang Li, Lijie Wen, and Philip S. Yu. 2022. Hiure: Hierarchical exemplar contrastive learning for unsupervised relation extraction. In *Proc. of NAACL*.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3219–3232, Brussels, Belgium. Association for Computational Linguistics.

Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. 2019. A general framework for information extraction using dynamic span graphs. In *Proceedings of NAACL*, pages 3036–3046, Minneapolis, Minnesota. Association for Computational Linguistics.

Lianbo Ma, Huimin Ren, and Xiliang Zhang. 2021. Effective cascade dual-decoder model for joint entity and relation extraction. *CoRR*, abs/2106.14163.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using LSTMs on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1116, Berlin, Germany. Association for Computational Linguistics.

Tapas Nayak and Hwee Tou Ng. 2020. Effective modeling of encoder-decoder architecture for joint entity and relation extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8528–8535.

Hao Peng, Tianyu Gao, Xu Han, Yankai Lin, Peng Li, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 2020. Learning from context or names? an empirical study on neural relation extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3661–3672.

Yujia Qin, Yankai Lin, Ryuichi Takanobu, Zhiyuan Liu, Peng Li, Heng Ji, Minlie Huang, Maosong Sun, and Jie Zhou. 2021. Erica: Improving entity and relation understanding for pre-trained language models via contrastive learning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3350–3363.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases, European Conference, ECML PKDD 2010, Barcelona, Spain, September 20-24, 2010, Proceedings, Part III*, volume 6323, pages 148–163. Springer.

Yongliang Shen, Xinyin Ma, Zeqi Tan, Shuai Zhang, Wen Wang, and Weiming Lu. 2021a. Locate and label: A two-stage identifier for nested named entity recognition. In *Proceedings of ACL*, pages 2782–2794.

Yongliang Shen, Xinyin Ma, Yechun Tang, and Weiming Lu. 2021b. A trigger-sense memory flow framework for joint entity and relation extraction. In *Proceedings of the Web Conference 2021*, WWW '21, page 1704–1715, New York, NY, USA. Association for Computing Machinery.

Yongliang Shen, Xiaobin Wang, Zeqi Tan, Guangwei Xu, Pengjun Xie, Fei Huang, Weiming Lu, and Yueting Zhuang. 2022. Parallel instance query network for named entity recognition. In *Proceedings of ACL*, pages 947–961.

Livio Baldini Soares, Nicholas Fitzgerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905.

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

Dianbo Sui, Yubo Chen, Kang Liu, Jun Zhao, Xiangrong Zeng, and Shengping Liu. 2020. Joint entity and relation extraction with set prediction networks. *arXiv preprint arXiv:2011.01675*.

Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, et al. 2020. Sparse r-cnn: End-to-end object detection with learnable proposals. *arXiv preprint arXiv:2011.12450*.

Zeqi Tan, Yongliang Shen, Shuai Zhang, Weiming Lu, and Yueting Zhuang. 2021. A sequence-to-set network for nested named entity recognition. In *Proceedings of IJCAI*, pages 3936–3942. International Joint Conferences on Artificial Intelligence Organization.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Jesse Vig. 2019. A multiscale visualization of attention in the transformer model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 37–42, Florence, Italy. Association for Computational Linguistics.

Christopher Walker, Stephanie Strassel, and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus. linguistic. In *Linguistic Data Consortium, Philadelphia 57*.

Haoyu Wang, Ming Tan, Mo Yu, Shiyu Chang, Dakuo Wang, Kun Xu, Xiaoxiao Guo, and Saloni Potdar. 2019. Extracting multiple-relations in one-pass with pre-trained transformers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1371–1377, Florence, Italy. Association for Computational Linguistics.

Jue Wang and Wei Lu. 2020. Two are better than one: Joint entity and relation extraction with table-sequence encoders. In *Proceedings of EMNLP 2020*, pages 1706–1721, Online. Association for Computational Linguistics.

Yijun Wang, Changzhi Sun, Yuanbin Wu, Junchi Yan, Peng Gao, and Guotong Xie. 2020a. Pre-training entity relation encoder with intra-span and inter-span information. In *Proceedings of EMNLP 2020*, pages 1692–1705, Online. Association for Computational Linguistics.

Yijun Wang, Changzhi Sun, Yuanbin Wu, Hao Zhou, Lei Li, and Junchi Yan. 2021. UniRE: A unified label space for entity relation extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 220–231, Online. Association for Computational Linguistics.

Yucheng Wang, Bowen Yu, Yueyang Zhang, Tingwen Liu, Hongsong Zhu, and Limin Sun. 2020b. TPLinker: Single-stage joint extraction of entities and relations through token pair linking. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1572–1582. International Committee on Computational Linguistics.

Zhepei Wei, Jianlin Su, Yue Wang, Yuan Tian, and Yi Chang. 2020. A novel cascade binary tagging framework for relational triple extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1476–1488. Association for Computational Linguistics.

Zhiheng Yan, Chong Zhang, Jinlan Fu, Qi Zhang, and Zhongyu Wei. 2021. A partition filter network for joint entity and relation extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 185–197, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hongbin Ye, Ningyu Zhang, Shumin Deng, Mosha Chen, Chuanqi Tan, Fei Huang, and Huajun Chen. 2021. Contrastive triple extraction with generative transformer. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Virtual Event, February 2-9, 2021*, pages 14257–14265. AAAI Press.

Yue Yuan, Xiaofei Zhou, Shirui Pan, Qiannan Zhu, Zeliang Song, and Li Guo. 2020. A relation-specific attention network for joint entity and relation extraction. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 4054–4060.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2335–2344, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Daojian Zeng, Haoran Zhang, and Qianying Liu. 2020. Copymtl: Copy mechanism for joint extraction of entities and relations with multi-task learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9507–9514.

Xiangrong Zeng, Daojian Zeng, Shizhu He, Kang Liu, and Jun Zhao. 2018. Extracting relational facts by an end-to-end neural model with copy mechanism. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1:*

Long Papers), pages 506–514, Melbourne, Australia. Association for Computational Linguistics.

Meishan Zhang, Yue Zhang, and Guohong Fu. 2017. End-to-end neural relation extraction with global optimization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1730–1740, Copenhagen, Denmark. Association for Computational Linguistics.

Shu Zhang, Dequan Zheng, Xinchen Hu, and Ming Yang. 2015. Bidirectional long short-term memory networks for relation classification. In *Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation*, pages 73–78, Shanghai, China.

Xin Zhang, Yong Jiang, Xiaobin Wang, Xuming Hu, Yueheng Sun, Pengjun Xie, and Meishan Zhang. 2022. Domain-specific NER via retrieving correlated samples. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2398–2404.

Xin Zhang, Guangwei Xu, Yueheng Sun, Meishan Zhang, and Pengjun Xie. 2021. Crowdsourcing learning as domain adaptation: A case study on named entity recognition. In *Proceedings of ACL*, pages 5558–5570.

Kang Zhao, Hua Xu, Yue Cheng, Xiaoteng Li, and Kai Gao. 2021. Representation iterative fusion based on heterogeneous graph neural network for joint entity and relation extraction. *Knowledge-Based Systems*, page 106888.

Tianyang Zhao, Zhao Yan, Yunbo Cao, and Zhoujun Li. 2020. Asking effective and diverse questions: A machine reading comprehension based framework for joint entity-relation extraction. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3948–3954. International Joint Conferences on Artificial Intelligence Organization. Main track.

Hengyi Zheng, Rui Wen, Xi Chen, Yifan Yang, Yunyan Zhang, Ziheng Zhang, Ningyu Zhang, Bin Qin, Xu Ming, and Yefeng Zheng. 2021. PRGC: potential relation and global correspondence based joint relational triple extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 6225–6235. Association for Computational Linguistics.

Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. 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 (Volume 1: Long Papers)*, pages 1227–1236, Vancouver, Canada. Association for Computational Linguistics.

Zexuan Zhong and Danqi Chen. 2021. A frustratingly easy approach for entity and relation extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 50–61, Online. Association for Computational Linguistics.

Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. 2020. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*.

## A Number of queries

Since the number of queries is pre-fixed, we conduct a comparison experiment in Table 6 to find the suitable setting. We employ the experiment on SciERC with a maximum number of 10 triples in one sentence. We observe that when the number is obviously larger than the ground truth, the performance of the model does decrease. We argue that this is due to the imbalance between positive and negative samples, as redundant queries predict the none category $\varnothing$. Eventually, the query number $M$ in our experiments is set to 15, which is deployed on all the other datasets.

| $M$ | NER | | | RE | | |
|---|---|---|---|---|---|---|
| | P | R | F | P | R | F |
| 10 | **68.2** | 70.0 | 69.1 | **40.1** | 37.8 | 38.9 |
| 15 | 68.1 | **71.2** | **69.6** | 39.7 | **38.9** | **39.3** |
| 20 | 67.1 | 70.6 | 68.8 | 39.1 | 37.7 | 38.4 |
| 30 | 66.9 | 70.2 | 68.5 | 38.2 | 36.8 | 37.5 |
| 50 | 68.1 | 70.3 | 69.2 | 38.6 | 36.1 | 37.3 |
| 100 | 66.2 | 70.5 | 68.3 | 39.9 | 35.9 | 37.8 |

Table 6: The performance impact of different number of queries on the SciERC development set.

| Inter-task | NER | | | RE | | |
|---|---|---|---|---|---|---|
| | P | R | F | P | R | F |
| Default | **68.1** | **71.2** | **69.6** | **39.7** | **38.9** | **39.3** |
| w/o ent-rel | 67.6 | 70.9 | 69.2 | 38.5 | 37.9 | 38.2 |
| w/o rel-ent | 67.5 | 70.4 | 68.9 | 39.5 | 38.3 | 38.9 |
| w/o both | 67.2 | 69.8 | 68.5 | 38.3 | 37.5 | 37.9 |

Table 7: The performance with different attention mask settings on the SciERC development set.

## B Analysis of query branches

In this section, we explore the entity and relation query branches in our decoder, the detailed architecture is shown in Figure 5. Based on different attention mask matrices for queries, there are three settings. (1) **w/o ent-rel**: entity queries are not visible to relation queries. (2) **w/o rel-ent**: relation queries are not visible to entity queries. (3) **w/o both**: entity queries and relation queries are not visible to each other. As shown in Table 7, if entity queries are visible to relation queries, the performance of the model on RE will improve by +1.1% (from 38.2% to 39.3%). Correspondingly, if relation queries are visible to entity queries, the model performance on NER will increase by +0.7% (from 68.9% to 69.6%). Further, if they are visible to each other, the F1-measure of the model on NER and RE will be improved by +1.1% and +1.4%, respectively. We attribute this to the fact that we can model the inherent dependencies between NER and RE tasks through the attention between entity queries and relation queries.

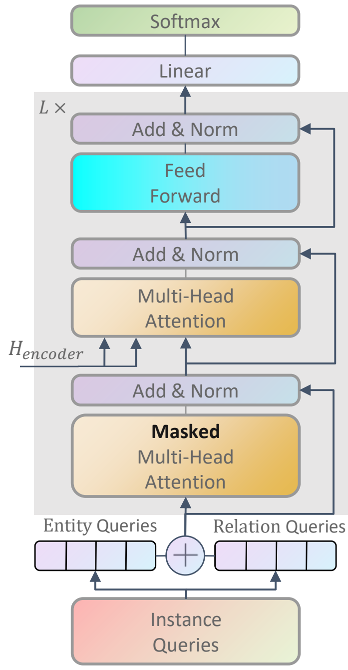


Figure 5: The entity and relation query branches with masked attention in decoder.

| Dateset | Normal | SEO | EPO | SOO |
|---|---|---|---|---|
| NYT | 3,266 | 1,297 | 978 | 45 |
| WebNLG | 239 | 448 | 6 | 85 |
| NYT* | 3,222 | 1,273 | 969 | 117 |

Table 8: Statistics on the test sets with different triple overlapping patterns.

| Dateset | N=1 | N=2 | N=3 | N=4 | N>5 |
|---|---|---|---|---|---|
| NYT | 3,244 | 1,045 | 312 | 291 | 108 |
| WebNLG | 256 | 175 | 138 | 93 | 41 |
| NYT* | 3,240 | 1,047 | 314 | 290 | 109 |

Table 9: Statistics on the test sets with different triple numbers in each sentence.

## C Case study

In order to provide a more intuitive understanding of the connection between queries, we visualize the attention weights between entity queries and between relation queries in the decoder with bertviz

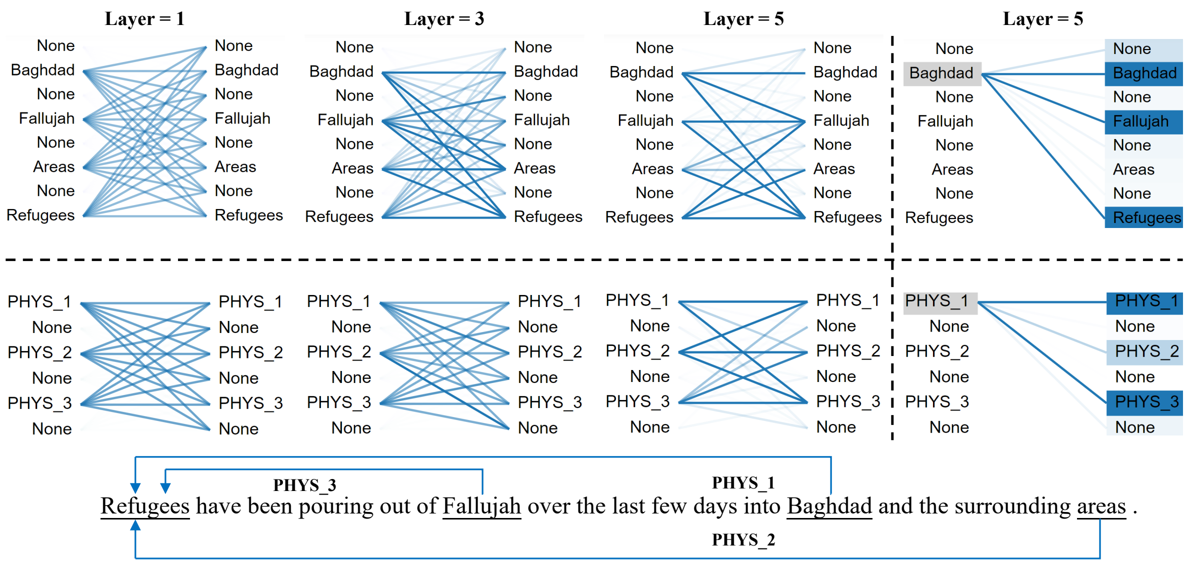


Figure 6: Visualization for the attention weights between entity queries and between relation queries. We randomly select "None" with the same number of gold labels, which indicates a non-entity or a non-relation corresponding to a query, and the attention of "None" is ignored. The sentence is randomly selected from the ACE05 corpus.

(Vig, 2019). First, we analyze the connection between the queries corresponding to the gold labels and the queries corresponding to the "None" labels, and then we analyze the connection between queries corresponding to the gold labels.

As shown in the left three columns in Figure 6, the attentions are weighted equally between queries before entering the decoding layers. However, as the decoding layer deepens, the queries corresponding to the gold labels will gradually neglect the none labels. For entity queries, when the number of decoding layers is 5, the entity queries corresponding to gold entities have most of their attention focused on themselves. The same holds true for relation queries. The observation suggests that these queries do capture dependencies between entities and between relations. And they are capable of considering the gold and none labels separately during decoding.

Beyond being able to distinguish between the gold and none labels, the queries corresponding to the gold labels have different attention weights from each other. From the right column of Figure 6, we can notice that the entity "Baghdad" pays more attention to "Refugees" and "Fallujah" but ignores "Areas" which is a `LOC` entity. This is due to the fact that "Baghdad" has the relation `PHYS_1` with "Refugees", and has the same entity type `GPE` as "Fallujah". Similarly, the relation `PHYS_1` shows more attention to `PHYS_3` than to `PHYS_2`, since `PHYS_1` and `PHYS_3` have the same subject entity "Refugees" and the same type `GPE` of object entity. It indicates that our approach can well exploit the dependencies between triples.