Computers, Materials & Continua
DOI: 10.32604/cmc.2023.032168
Article

Tech Science Press

A Two-Phase Paradigm for Joint Entity-Relation Extraction

Bin Ji[1], Hao Xu[1], Jie Yu[1], Shasha Li[1], Jun Ma[1], Yuke Ji[2,*] and Huijun Liu[1]

[1]College of Computer, National University of Defense Technology, Changsha, 410073, China
[2]The Affiliated Eye Hospital of Nanjing Medical University, Nanjing, 210029, China
*Corresponding Author: Yuke Ji. Email: jiyuke93@163.com

Received: 09 May 2022; Accepted: 12 June 2022

Abstract: An exhaustive study has been conducted to investigate span-based models for the joint entity and relation extraction task. However, these models sample a large number of negative entities and negative relations during the model training, which are essential but result in grossly imbalanced data distributions and in turn cause suboptimal model performance. In order to address the above issues, we propose a two-phase paradigm for the span-based joint entity and relation extraction, which involves classifying the entities and relations in the first phase, and predicting the types of these entities and relations in the second phase. The two-phase paradigm enables our model to significantly reduce the data distribution gap, including the gap between negative entities and other entities, as well as the gap between negative relations and other relations. In addition, we make the first attempt at combining entity type and entity distance as global features, which has proven effective, especially for the relation extraction. Experimental results on several datasets demonstrate that the span-based joint extraction model augmented with the two-phase paradigm and the global features consistently outperforms previous state-of-the-art span-based models for the joint extraction task, establishing a new standard benchmark. Qualitative and quantitative analyses further validate the effectiveness the proposed paradigm and the global features.

Keywords: Joint extraction; span-based; named entity recognition; relation extraction; data distribution; global features

1 Introduction

Span-based joint entity and relation extraction models simultaneously conduct NER (**N**amed **E**ntity **R**ecognition) and RE (**R**elation **E**xtraction) in text span forms. Typically, these models are constructed as follows: given an unstructured text, the model divides it into text spans; it then constructs ordered span pairs (a.k.a. relation tuples); and finally, it obtains entities and relations by performing classifications on the semantic representations of spans and relation tuples, respectively. We present a typical case study in Fig. 1: the "In", "In 1831", and "James Garfield" are three span examples; the <"James Garfield", "U.S."> and <"James Garfield", "Ohio"> are two relation tuple examples; a span-based model predicts the types of spans and relation tuples by performing classifications on related semantic representations. For instance, the "In" is classed as the Not-Entity type, and the <"James Garfield", "Ohio"> is classified as the Live type.[1]

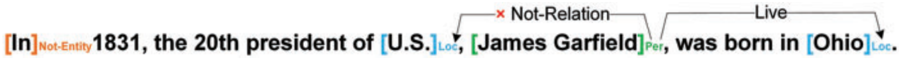

[In]Not-Entity 1831, the 20th president of [U.S.]Loc, [James Garfield]Per, was born in [Ohio]Loc.
× Not-Relation
Live

Figure 1: An example of the span-based joint extraction. The Loc and Per are two pre-defined entity types, and the Live is a pre-defined relation type. × denotes the <"James Garfield", "U.S.">, supposedly classified into the Live type, is actually classified into the Not-Relation type by SpERT [1], a Span-based joint Entity-Relation extraction model with Transformer

Span-based joint extraction models [2–7] sample numerous negative entities and relations (i.e., spans of the Not-Entity type and relation tuples of the Not-Relation type) during the model training. These negative examples actually lead to grossly imbalanced data distributions, which is one of the primary reasons for the suboptimal model performance. As shown in Tab. 1, the entity distribution between Other and Not-Entity is 592: 101555 (approximate to 1: 172), the relation distribution between Kill and Not-Relation is 229: 12915 (approximate to 1: 56). Paradoxically, previous work [1] demonstrates that an adequate number of negative examples are required to ensure that the model performs well. Thus, resolving the issue of grossly imbalanced data distributions while maintaining an adequate number of negative examples is a feasible way to improve the model performance.

Table 1: The entity and relation counts in CoNLL04, which are obtained using SpERT. The Other, Org, Per and Loc are four pre-defined entity types, and the Kill, LocIn, Work, OrgBI and Live are five pre-defined relation types, and the Not-Entity and Not-Relation are the types of negative entities and relations, respectively

NER	Other	Org	Per	Loc		Not-Entity
	592	786	1,370	1,541		101,555
RE	Kill	LocIn	Work	OrgBI	Live	Not-Relation
	229	312	325	317	421	12,915

Global features, such as those derived from entity information, can be critical in the joint extraction task. As illustrated in Fig. 1, if SpERT is aware that the "James Garfield" is a person (Per) entity and the "U.S." is a location (Loc) entity beforehand, it may easily classify the <"James Garfield", "U.S."> into the Live type. Moreover, entity distance, which tracks the word counts between two entities, can reflect the entities' correlation. For example, in the CoNLL04 dataset, relations with an entity distance of less than 6 account for 64.5%, and the smaller the distance, the more likely the two entities have a relation. However, as far as we know, previous work [8–12] has used either entity type or entity distance but not both. The combination of the above two types of information may play a more important role in the joint extraction task. As shown in Tab. 2, the <Loc, Loc> tends to have the LocIn relation when the entity distance is smaller, such as 76.6% for [0–3], 12.8% for [4–7] and 3.5% for [8–11], whereas the <Per, Per> tends to have the Kill relation in the case of a bigger entity distance, such as 21.3% for [0–3], 33.5% for [4–7], and 26.7% for [8–11].

[1] Span-based models add a Not-Entity type for spans that are not entities and a Not-Relation type for relation tuples that don't hold relations.

Table 2: Entity distance statistics of the CoNLL04 dataset. We use the ordered entity type tuple to denote ordered relation tuples of the same type, such as the <Per, Loc> denotes all relation tuples that the type of their first entity is the Per and the type of their second entity is the Loc. We divide all distances into four distance intervals, i.e., [0–3], [4–7], [8–11] and [>11]

Relation tuple	Type	Entity distance			
		[0–3]	[4–7]	[8–11]	[>11]
<Per, Loc>	Live	39.0%	27.6%	10.5%	22.9%
<Loc, Loc>	LocIn	76.6%	12.8%	3.5%	7.1%
<Per, Org>	Work	41.5%	35.1%	8.0%	15.4%
<Org, Loc>	OrgBI	72.0%	10.7%	5.5%	11.8%
<Per, Per>	Kill	21.3%	33.5%	26.7%	18.5%

In this paper, we propose a two-phase span-based model for the joint extraction task, with the goal of addressing the issue of grossly imbalanced data distributions and the lack of effective global features. Motivated by the fact that we can achieve NER (RE) in two steps, namely first classify all entities (relations) and then predict their types. We divide the joint extraction task into two phases, with the first phase obtaining entities and relations and the second phase predicting their types. Our model reduces the data distribution gap by dozens of times using the two-phase paradigm. Take the data in Tab. 1 as an example: (1) in the first stage, the entity distribution can be reduced to 1: 24 and the relation distribution to 1: 8, whereas the corresponding values in SpERT are 1: 172 and 1: 56, respectively.[2] (2) In the second phase, our model predicts the types of entities and relations, implying that the data distributions are roughly even.[3] Moreover, we attempt for the first time to combine entity type and entity distance as global features and use them to augment our model. Furthermore, we propose a gated mechanism for fusing various semantic representations, taking the weighted importance of each representation into account. In Section 4.5, we validate the effectiveness of the above model components.

Experimental results on the ACE05, CoNLL04 and SciERC datasets demonstrate that our model consistently outperforms the strongest span-based baselines in terms of F1-score, providing a new span-based benchmark for the joint extraction task. Extensive analyses further validate the effectiveness of our model.

In summary, our model differs from the previous span-based models in three ways: (1) As far as we know, our model makes the first attempt to balance the grossly imbalanced data distributions. (2) Our model combines entity type and entity distance as the global features, whereas previous span-based models use at most one of them. (3) Our model uses a gated mechanism to fuse various semantic representations, whereas previous span-based models use a simple concatenation manner.

[2] 1: 24 ≈ (592+786+1370+1541): 101555; 1: 8 ≈ (229+312+325+347+421): 12915.

[3] The entity distributions can be approximate to (592: 786: 1370: 1541), and the relation distributions can be approximate to (229: 312: 325: 347: 421), which are approximately even.

2 Related Work

2.1 Span-based Joint Entity and Relation Extraction

Recently, span-based models have been extensively investigated for the joint entity and relation extraction task. Luan et al. [2] propose almost the first span-based joint model and attempt to further improve model performance by incorporating the coreference resolution task [13,14]. Luan et al. [4] also include the coreference resolution task in their span-based joint model. Moreover, some other span-based models [5] have examined how to incorporate additional natural language processing tasks, such as event detection [15,16]. More recently, Dixit and Al-Onaizan [3] introduce the pre-trained language model, i.e., ELMo (**E**mbeddings from **L**anguage **Mo**dels) [17], into the span-based joint model for the first time. Eberts and Ulges [1] propose to use BERT (**B**idirectional **E**ncoder **R**epresentation from **T**ransformers) [18] as the backbone of their span-based joint model. Zhong and Chen [7] propose to use ALBERT (**A** **L**ite **BERT**) [19] in their span-based joint model. However, these models suffer from grossly imbalanced data distributions, as the span-based paradigm requires extensive negative entities and relations. Although our model also samples a large number of examples, we propose a two-phase paradigm to eliminate the data distribution gap effectively.

2.2 Global Features

The entity type and entity distance are two types of important global features that are frequently used in joint extraction models [20–27]. Miwa and Bansal [18], Sun and Grishman [28], and Bekoulis et al. [9], are among the first to use entity types as global features in their joint extraction models. They concatenate fixed-size embeddings trained for entity types to relation semantic representations. Zhao et al. [10] model strong correlations between entity labels and text tokens and concatenate entity label embeddings to relation semantic representations. For entity distance, Zeng et al. [11] and Ye et al. [12] concatenate relative entity position features to relation semantic representations. However, the above models use either entity type or entity distance but make no attempt to combine them. In comparison, our model suggests combining the entity type and entity distance as global features, which is validated to be more effective.

3 Model

The neural architecture of our two-phase span-based model is illustrated in Fig. 2. For a given unstructured text $\mathcal{T} = (t_1, t_2, \ldots, t_n)$ where t_i denotes the *i-th* text token, our model first obtains its BERT embedding sequence (Section 3.1); then in Phase One, our model obtain entities and relations by performing binary classifications on semantic representations of spans and relation tuples, respectively. These entities and relations are referred to as coarse-grained entities and relations, respectively (Section 3.2); next, in Phase Two, our model predicts the types of these coarse-grained entities and relations, obtaining fine-grained entities and relations (Section 3.3). In both phases, we combine the entity type and entity distance as global features and use a gated mechanism to fuse various semantic representations.

We formulate the text spans (denoted as $\mathcal{S}$) from $\mathcal{T}$ as follows:

$$\mathcal{S} = \left(t_j,\ t_{j+1}, \ldots,\ t_{j+k}\right) \quad s.t. \quad 0 \le j \le j+k \le n \text{ and } k \le \epsilon, \tag{1}$$

where ϵ is the span width threshold.

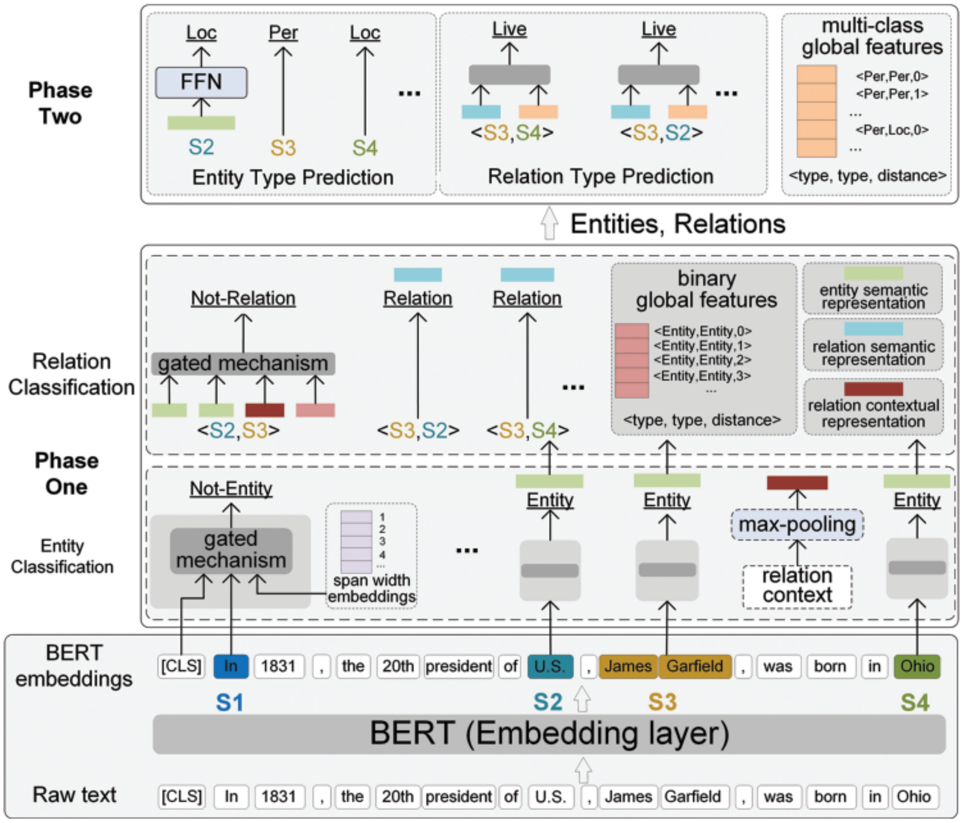


Figure 2: Neural architecture of the proposed model. In the Phase One, the model classifies entities and relations; in the phase Two, the model predicts their types. In both phases, the model combines entity type and distance as global features

3.1 Embedding Layer

Our model uses the BERT [18] model as the word embedding generator. We denote the BERT embedding sequence for text $\mathcal{T}$ as follows:

$$\boldsymbol{E}_{\mathcal{T}} = (\boldsymbol{x}_0,\ \boldsymbol{x}_1,\ \boldsymbol{x}_2, \ldots,\ \boldsymbol{x}_n), \tag{2}$$

where $\boldsymbol{E}_{\mathcal{T}} \in \mathbb{R}^{(n+1)*d}$ and d is the BERT embedding dimension. $\boldsymbol{x}_0$ is the BERT embedding for the added [CLS] token, which is a built-in setting of the BERT model.[4] $\boldsymbol{x}_i$ is the BERT embedding for the token t_i. Due to the fact that BERT may tokenize a token into several sub-tokens in order to avoid the Out-of-Vocabulary (OOV) problem, we obtain $\boldsymbol{x}_i$ by applying the max-pooling function to the BERT embeddings of the sub-tokens tokenized from the token t_i.

Based on $\boldsymbol{E}_{\mathcal{T}}$, We denote the BERT embedding sequence for span $\mathcal{S}$ as follows:

$$\boldsymbol{E}_{\mathcal{S}} = \left(\boldsymbol{x}_j,\ \boldsymbol{x}_{j+1}, , \ldots,\ \boldsymbol{x}_{j+k}\right). \tag{3}$$

[4] The [CLS] token is a specific token that added to the beginning of tokenized texts. The embedding of the [CLS] token is generally used for text classifications.

3.2 Phase One

As shown in Fig. 2, the Phase One is composed of two modules: **Entity Classification** and **Relation Classification**, where the former obtains coarse-grained entities and the latter obtains coarse-grained relations.

3.2.1 Entity Classification

This module obtains coarse-grained entities by performing binary classification on span semantic representations. We begin by converting all entity types in the training set to the Entity type and set the type of sampled negative entities to the Not-Entity type. Our model will be trained to classify spans as the Entity type when they are predicted to be entities, otherwise the Not-Entity type.

In this paper, we obtain the span semantic representations using three different types of semantic representations: (1) span token representation, (2) contextual representation, and (3) span width embedding.

For the span $\mathcal{S}$, we obtain its token representation (denoted as $\hat{\boldsymbol{E}}_{\mathcal{S}}$) by applying the max-pooling function to its BERT embedding sequence $\boldsymbol{E}_{\mathcal{S}}$:

$$\hat{\boldsymbol{E}}_{\mathcal{S}} = \left[\max\left(\boldsymbol{x}_{j,1},\ \boldsymbol{x}_{j+1,1}, , \ldots,\ \boldsymbol{x}_{j+k,1}\right),\ \max\left(\boldsymbol{x}_{j,2},\ \boldsymbol{x}_{j+1,2}, , \ldots,\ \boldsymbol{x}_{j+k,2}\right),\ \ldots,\ \max\left(\boldsymbol{x}_{j,d},\ \boldsymbol{x}_{j+1,d}, , \ldots,\ \boldsymbol{x}_{j+k,d}\right)\right], \tag{4}$$

where $\hat{\boldsymbol{E}}_{\mathcal{S}} \in \mathbb{R}^d$.

In this paper, we take the $\boldsymbol{x}_0 \in \mathbb{R}^d$ as the contextual representation for any span $\mathcal{S}$ from the text $\mathcal{T}$.

Span width embedding allows the model to incorporate prior experience over span widths. In this paper, we train a fixed-size embedding for each span width (i.e., 1, 2, ...) during the model training. And we refer to the width embedding for the span $\mathcal{S}$ (length is $j+1$) as $\boldsymbol{W}_{j+1}$, where $\boldsymbol{W}_{j+1} \in \mathbb{R}^d$.

$\hat{\boldsymbol{E}}_{\mathcal{S}}$, on the other hand, should theoretically contribute the most to the span semantic representation, whereas $\boldsymbol{W}_{j+1}$ the least. However, the previous work [1,4] has overlooked this critical property, concatenating the above representations, which has been demonstrated to be insufficient [10]. In this paper, we propose a gated mechanism that enables us to weigh the importance of each representation. The span semantic representation (denoted as $\boldsymbol{E}'_{\mathcal{S}}$) is then obtained by summing the weighted representations:

$$v_o = \boldsymbol{W}_1\boldsymbol{E}_o + b_1 \quad s.t. \quad 1 \leq o \leq \delta, \tag{5a}$$

$$\alpha_o = \frac{\exp^{v_o}}{\sum_{m=1}^{\delta} \exp^{v_m}}, \tag{5b}$$

$$\boldsymbol{E}'_{\mathcal{S}} = \sum_{m=1}^{\delta} \alpha_m \boldsymbol{E}_m, \tag{5c}$$

where $\boldsymbol{W}_1 \in \mathbb{R}^d$, b_1 is a scalar and $\left\{\boldsymbol{E}_o,\ \boldsymbol{E}'_{\mathcal{S}}\right\} \in \mathbb{R}^d$. In the current scenario, the δ is set to 3, and $\boldsymbol{E}_1$, $\boldsymbol{E}_2$, and $\boldsymbol{E}_3$ are $\hat{\boldsymbol{E}}_{\mathcal{S}}$, $\boldsymbol{x}_0$, and $\boldsymbol{W}_{j+1}$, respectively.

To obtain coarse-grained entities, we first pass the $\boldsymbol{E}'_{\mathcal{S}}$ through an FFN (**F**eed **F**orward **N**etwork), and then feed it into the sigmoid function, which yields probability distributions for the span $\mathcal{S}$ on the above two types, i.e., Entity and Not-Entity:

$$\hat{\boldsymbol{E}}'_{\mathcal{S}} = \boldsymbol{W}_2\boldsymbol{E}'_{\mathcal{S}} + \boldsymbol{b}_2, \tag{6a}$$

$$\boldsymbol{y}_{\mathcal{S},i} = \frac{1}{1 + \exp^{\hat{\boldsymbol{E}}'_{\mathcal{S},i}}}, \tag{6b}$$

where $\boldsymbol{W}_2 \in \mathbb{R}^{2*d}$ and $\boldsymbol{b}_2 \in \mathbb{R}^2$ are trainable FFN parameters, $\hat{\boldsymbol{E}}'_{\mathcal{S}} \in \mathbb{R}^2$. By searching the highest-scored class, $\boldsymbol{y}_{\mathcal{S}}$ estimates whether $\mathcal{S}$ is a coarse-grained entity or not. We build a coarse-grained entity set T_e with the predicted entities.

3.2.2 Relation Classification

This module obtains coarse-grained relations by performing binary classification on semantic representations of relation tuples. We begin by converting all relation types in the training set to the Relation type and assigning the Not-Relation type to sampled negative relations. Our model will be trained to classify relation tuples as the Relation type if they have relations, otherwise the Not-Relation type.

Let e_2 and e_2 be any two coarse-grained entities of T_e. We formulate relation tuples as follows:

$$r_b = \langle e_1,\ e_2 \rangle \quad s.t. \quad e_1,\ e_2 \in T_e \text{ and } e_1 \neq e_2. \tag{7}$$

We obtain the sematic representation of r_b with four different types of semantic representations, namely (1) the representation of e_1, (2) the representation of e_2, (3) relation contextual representation, and (4) global features. We use $\boldsymbol{E}'_{e_1}$ and $\boldsymbol{E}'_{e_2}$, which are calculated using the Eq. (5) in Section 3.2.1, as the representations of e_1 and e_2, respectively.

Relation context is the text that between the two entities of a relation tuple [29]. In this paper, we assume the relation context of r_b as $Con = \left(\boldsymbol{t}_p,\ \boldsymbol{t}_{p+1}, \ldots,\ \boldsymbol{t}_{p+q}\right)$. Thus the BERT embedding sequence for Con is as follows:

$$\boldsymbol{E}_c = \left(\boldsymbol{x}_p,\ \boldsymbol{x}_{p+1}, \ldots,\ \boldsymbol{x}_{p+q}\right). \tag{8}$$

We obtain the contextual representation of r_b (denoted as $\boldsymbol{R}_c$) by applying the max-pooling function to $\boldsymbol{E}_c$:

$$\boldsymbol{R}_C = \left[\max\left(\boldsymbol{x}_{p,1},\ \boldsymbol{x}_{p+1,1}, \ldots,\ \boldsymbol{x}_{p+q,1}\right),\ \max\left(\boldsymbol{x}_{p,2},\ \boldsymbol{x}_{p+1,2}, \ldots,\ \boldsymbol{x}_{p+q,2}\right),\ \ldots,\ \max\left(\boldsymbol{x}_{p,d},\ \boldsymbol{x}_{p+1,d}, \ldots,\ \boldsymbol{x}_{p+q,d}\right)\right]. \tag{9}$$

In this paper, we propose to combine entity type and entity distance as global features. Due to the fact that all entities here are the Entity type, only the entity distance can be used to distinguish different feature entries. As show in Fig. 2, we refer to them as binary global features. During model training, we train a fixed-size embedding for each feature entry and denote the feature embedding for r_b as $\boldsymbol{D}_{r_b}$, where $\boldsymbol{D}_{r_b} \in \mathbb{R}^d$.

We obtain the semantic representation of r_b (denoted as $\boldsymbol{R}_{r_b}$) using the proposed gated mechanism, as shown in Eq. (5). In the current scenario, the δ is set to 4, and $\boldsymbol{E}_1$, $\boldsymbol{E}_2$, $\boldsymbol{E}_3$, and $\boldsymbol{E}_4$ are $\boldsymbol{E}'_{e_1}$, $\boldsymbol{E}'_{e_2}$, $\boldsymbol{R}_C$, and $\boldsymbol{D}_{r_b}$, respectively.

To obtain coarse-grained relations, we first pass the $\boldsymbol{R}_{r_b}$ through an FFN and then feed it into the sigmoid function, which yields probability distributions for r_b on the above two types, i.e., Relation and Not-Relation:

$$\hat{\boldsymbol{R}}_{r_b} = \boldsymbol{W}_3\boldsymbol{R}_{r_b} + \boldsymbol{b}_3, \tag{10a}$$

$$\boldsymbol{y}_{r_b,i} = \frac{1}{1 + \exp^{\hat{\boldsymbol{R}}_{r_b,i}}}, \tag{10b}$$

where $\boldsymbol{W}_3 \in \mathbb{R}^{2*d}$ and $\boldsymbol{b}_3 \in \mathbb{R}^2$ are trainable FFN parameters, and $\hat{\boldsymbol{R}}_{r_b} \in \mathbb{R}^2$. By searching the highest-scored class, $\boldsymbol{y}_{r_b}$ estimates whether r_b has a relation or not. We build a coarse-grained relation set T_r with the predicted relations.

3.2.3 Training Loss of Phase One

For each of the above two binary classifications, the training objective is to minimize the following binary cross-entropy loss:

$$\mathcal{L}_b^t = -\frac{1}{N^t}\sum_{i=1}^{N^t} \left(\boldsymbol{y}_i^t \log \hat{\boldsymbol{y}}_i^t + \left(1 - \boldsymbol{y}_i^t\right)\left(1 - \hat{\boldsymbol{y}}_i^t\right)\right), \tag{11}$$

where t denotes one of the above two classifications. $\boldsymbol{y}_i^t$ is the one-hot vector of gold type. $\hat{\boldsymbol{y}}_i^t$ is the predicted probability distributions. N^t is the number of instances for the classification t.

3.3 Phase Two

In the Phase Two, our model predicts the types of coarse-grained entities and relations, obtaining fine-grained entities and relations. The Phase Two, as illustrated in Fig. 2, is composed of two modules: **Entity Type Predication** and **Relation Type Predication**.

3.3.1 Entity Type Predication

In this module, we obtain entity types by conducting multi-class classifications on the semantic representations of coarse grained entities. Specifically, for each coarse-grained entity e in T_e, we denote its semantic representation as $\boldsymbol{E}'_e \in \mathbb{R}^d$, which is obtained the same as the span semantic representation, as illustrated in Section 3.2.1. To obtain the type of e, we first pass $\boldsymbol{E}'_e$ through an FFN and then feed it into the softmax function, which yields probability distributions for e on Ω, where Ω is the set of all pre-defined entity types:

$$\hat{\boldsymbol{E}}'_e = \boldsymbol{W}_4\boldsymbol{E}'_e + \boldsymbol{b}_4, \tag{12a}$$

$$\hat{\boldsymbol{y}}_{e,i} = \frac{1}{\sum_{j=1}^{|\Omega|} \exp^{\hat{E}'_{e,j}}} \quad s.t. \quad 1 \le i \le |\Omega|, \tag{12b}$$

where $\boldsymbol{W}_4 \in \mathbb{R}^{|\Omega|*d}$ and $\boldsymbol{b}_4 \in \mathbb{R}^{|\Omega|}$ are trainable FFN parameters. $|\Omega|$ is the counts of pre-defined entity types. By searching the highest-scored class, $\hat{\boldsymbol{y}}_e$ estimates a pre-defined entity type for e.

3.3.2 Relation Type Predication

We obtain relation types by performing multi-class classifications on relation semantic representations. As shown in Fig. 2, the relation semantic representation is derived from two parts: the relation representation used for the binary relation classification and multi-class global features.

For each coarse-grained relation r in T_r, we denote its representation used for the binary relation classification as $\boldsymbol{R}_r$, which can be obtained using the same approach illustrated in Section 3.2.2. As shown in Fig. 2, we combine the entity type and entity distance as the multi-class global features. We formulate the combination of entity type and entity distance as follows:

$$C = \{\Omega \otimes \Omega \otimes \Delta\}, \tag{13}$$

where Ω is the set of pre-defined entity types, Δ is the set of entity distances, and $\otimes$ denotes the Cartesian Product. For each feature entry in C, we train a fixed-size embedding for it during the model training. We denote the feature embedding for r as $\boldsymbol{C}_r \in \mathbb{R}^d$.

Then we obtain the relation semantic representation (denoted as $\boldsymbol{R}'_r$) with $\boldsymbol{R}_r$ and $\boldsymbol{C}_r$, which is calculated using the Eq. (5). In the current scenario, the δ is set to 2, and $\boldsymbol{E}_1$ and $\boldsymbol{E}_2$ are $\boldsymbol{R}_r$ and $\boldsymbol{C}_r$, respectively.

To obtain the type of r, we first pass $\boldsymbol{R}'_r$ through an FFN, and then feed it into the softmax function, which yields probability distributions for r on Ψ, where Ψ is the set of all pre-defined relation types:

$$\hat{\boldsymbol{R}}'_r = \boldsymbol{W}_5 \boldsymbol{R}'_r + \boldsymbol{b}_5, \tag{14a}$$

$$\hat{\boldsymbol{y}}_{r,i} = \frac{\exp^{\hat{\boldsymbol{R}}'_{r,i}}}{\sum_{j=1}^{|\Psi|} \exp^{\hat{\boldsymbol{R}}'_{r,j}}} \quad s.t. \quad 1 \leq j \leq |\Psi|, \tag{14b}$$

where $\boldsymbol{W}_5 \in \mathbb{R}^{|\Psi| * d}$ and $\boldsymbol{b}_5 \in \mathbb{R}^{|\Psi|}$. $|\Psi|$ is the counts of pre-defined relation types. By searching the highest-scored class, $\hat{\boldsymbol{y}}_r$ estimates the type that r has.

3.3.3 Training Loss of the Phase Two

For each of above two multi-class classification tasks, the training objective is to minimize the following cross-entropy loss:

$$\mathcal{L}_p^{t'} = -\frac{1}{M^{t'}} \sum_{i=1}^{M^{t'}} \boldsymbol{y}_i^{t'} \log \hat{\boldsymbol{y}}_i^{t'}, \tag{15}$$

where t' denotes one of the above two classifications. $\boldsymbol{y}_i^{t'}$ is the one-hot vector of gold type. $\hat{\boldsymbol{y}}_i^{t'}$ is the predicted probability distributions. $M^{t'}$ is the number of instances for the classification t'.

3.4 Model Training

During the model training, we minimize the following joint training loss:

$$\mathcal{L}(\boldsymbol{W}; \boldsymbol{\theta}) = \sum_{t \in T} \mathcal{L}_b^t + \sum_{t' \in T'} \mathcal{L}_p^{t'}, \tag{16}$$

where T denotes the two binary classifications and T' denotes the two multi-class classifications.

4 Experiments

4.1 Dataset

We evaluate our model on ACE05 [30], CoNLL04 [31], and SciERC [2].

ACE05 defines seven entity types (Per, Org, Loc, Gpe, Fac, Veh, and Wea) and six relation types (Phys, Part-whole, Per-soc, Org-aff, Art, and Gen-aff) between entities. We use the same data splits, pre-processing, and task settings proposed by Li and Ji [32] and Li et al. [33]. It has 351 documents for training, 80 for development and 80 for test.

CoNLL04 defines four entity types (Loc, Org, Per, and Other) and five relation types (Kill, Live, LocIn, OrgBI, and Work). We use the splits defined by Ji et al. [6] and Wang et al. [25]. The dataset consists of 910 instances for training, 243 for development and 288 for test.

SciERC is derived from 500 abstracts of AI papers. The dataset defines six scientific entities (Task, Method, Metric, Material, Other, and Generic) and seven relation types (Compare, Conjunction, Evaluate-for, Used-for, Feature-of, Part-of, and Hyponym-of) in a total of 2,687 sentences. We use the same training (1,861 sentences), development (275 sentences), and test (551 sentences) split following the previous work [3,34].

4.2 Experimental Setup

For a fair comparison with previous work, we use the bert-base-cased model on ACE05 and CoNLL04, and use the scibert-scivocab-cased model on SciERC. We optimize our model using the BertAdam for 120 epochs with a learning rate of 5e-5 and a weight decay of 1e-2. We set the span width threshold ϵ to 10 for all datasets and the entity distance set Δ to $\{0, 1, \ldots, 10\}$, and if an entity distance is greater than 10, we set it to 10. Moreover, we employ the same negative sampling strategy proposed by Eberts and Ulges [1]. We use the standard Precision (P), Recall (R) and F1-score to evaluate the model performance:

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \tag{17a}$$

$$\mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \tag{17b}$$

$$\mathrm{F1} = \frac{2 * \mathrm{P} * \mathrm{R}}{\mathrm{P} + \mathrm{R}}, \tag{17c}$$

where TP, FP and FN stand for true positive, false positive, and false negative, respectively.

For ACE05, an entity mention is considered correct if its head region and type match the ground truth, and a relation is correct if both its relation type and two entity mentions are correct. For CoNLL04, an entity mention is considered correct if its offsets and type match the ground truth, and a relation is correct if both its relation type and two entity mentions are correct. For SciERC, the entity type is not considered when evaluating relation extraction, which is in line with the previous work [6,7]. And the remaining settings are identical to those for CoNLL04.

4.3 Main Results

We compare our model with all the published span-based models for the joint extraction task that we are aware of. We report the comparison results in Tab. 3–Tab. 5, from which we can observe that our model consistently outperforms the strongest baselines in terms of F1-score across the three datasets.

To be more precise, on ACE05, our model achieves + 0.4% and + 3.2% absolute F1 gains on NER and RE, respectively, when compared to Ji et al. [6] that achieves the previous best NER performance. In addition, when compared to Zhong and Chen [7] that achieves the previous best RE performance, our model achieves + 1.3% and + 1.4% absolute F1 gains on NER and RE, respectively. On CoNLL04, our model achieves + 0.3% and + 1.6% absolute F1 gains on NER and RE, respectively, when compared to the strongest baseline Ji et al. [6]. On SciERC, when compared to Santosh et al. [35] that achieves the previous best NER performance, our model delivers + 0.5% and + 1.4% absolute F1 gains. When compared to Zhang et al. [36] that achieves the previous best RE results, our model achieves + 0.6% and + 0.2% absolute F1 gains.

Table 3: Performance comparisons on ACE05. ∗ denotes using the bert-based-cased model and the single-sentence setting for a fair comparison. The bold values denote the best results

Model	NER			RE		
	P	R	F1	P	R	F1
Dixit and Al-Onaizan [3]	85.9	86.1	86.0	68.0	58.4	62.8
Luan et al. [4]	-	-	88.4	-	-	63.2
Wadden et al. [5]	-	-	88.6	-	-	63.4
Zhong and Chen [7] ∗	-	-	88.7	-	-	66.7
Wang et al. [37]	-	-	88.9	-	-	64.3
Ji et al. [6]	89.3	89.9	89.6	71.2	60.2	65.2
Our Model	**89.4**	**90.6**	**90.0**	**72.8**	**64.5**	**68.4**

Table 4: Performance comparisons on CoNLL04. The bold values denote the best results

Model	NER			RE		
	P	R	F1	P	R	F1
Eberts and Ulges [1]	88.3	89.6	89.0	73.0	70.0	71.5
Zhang et al. [36]	88.1	90.6	89.3	75.4	71.1	73.2
Tang et al. [38]	-	-	89.4	-	-	72.6
Ji et al. [6]	**90.1**	90.4	90.2	77.0	71.9	74.3
Our Model	89.6	**91.4**	**90.5**	**78.7**	**73.3**	**75.9**

Table 5: Performance comparisons on SciERC. ∗ denotes using the scibert-scivocab-cased model. The bold values denote the best results

Model	NER			RE		
	P	R	F1	P	R	F1
Luan et al. [2]	67.2	61.5	64.2	47.6	33.5	39.3
Luan et al. [4]	-	-	65.2	-	-	41.6
Wadden et al. [5]	-	-	67.5	-	-	48.4
Zhong and Chen [7] ∗	-	-	68.9	-	-	50.1
Eberts and Ulges [1] ∗	70.9	69.8	70.3	53.4	48.5	50.8
Zhang et al. [36] ∗	69.7	71.1	70.4	**55.3**	50.0	52.5
Santosh et al. [35] ∗	**69.8**	71.3	70.5	51.9	50.6	51.3
Our Model ∗	69.7	**72.3**	**71.0**	52.9	**52.5**	**52.7**

We attribute the above performance improvements to that our model is capable of balancing the grossly imbalanced data distributions and exploiting the effective global features.

4.4 Effectiveness Investigations

We conduct extensive effectiveness investigations across the three datasets and use SpERT [1] as the baseline. SpERT is the most similar model to ours, and it uses two linear decoders for entity and relation classifications, as well as the BERT model as a backbone. SpERT, on the other hand, ignores the global features and does not balance the imbalanced data distributions. Furthermore, to make a fair comparison, our model employs the same negative sampling strategy as SpERT.

4.4.1 Data Distributions

As illustrated in Tab. 6, we compare our model with the baseline in terms of the most imbalanced data distributions. We obtain the data distributions on NER and RE by comparing the numbers of different types of entities and relations, i.e., the smallest number *vs* the largest number. And we obtain the data distributions during the model training. We have the following observations: (1) On ACE05, the most imbalanced data distributions of the baseline are 1: 773.3 on NER and 1: 150.0 on RE. Our model, on the other hand, reduces the ratios to 1: 21.3 and 1: 13.8, respectively. (2) On CoNLL04, the most imbalanced data distributions of the baseline are 1: 171.5 on NER and 1: 56.4 on RE. Our model, on the other hand, reduces the ratios to 1: 23.7 and 1: 9.9, respectively. (3) On SciERC, the most imbalanced data distributions of the baseline are 1: 605.3 on NER and 1: 913.5 on RE. Our model, on the other hand, reduces the ratios to 1: 25.5 and 1: 35.6, respectively.

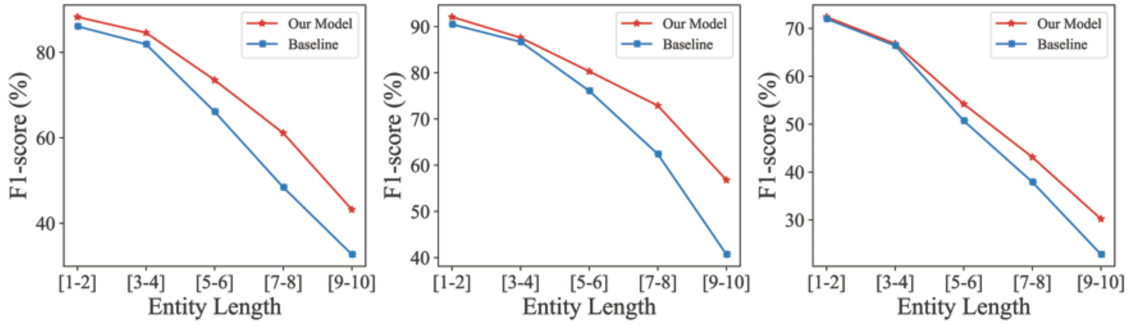
Table 6: Comparisons regarding the most imbalanced data distributions between our model and the baseline

Data	Task	Baseline	Our model	
			Phase one	Phase two
ACE05	NER	1: 773.3	1: 19.7	1: 21.3
	RE	1: 150.0	1: 13.8	1: 3.4
CoNLL04	NER	1: 171.5	1: 23.7	1: 2.6
	RE	1: 56.4	1: 9.9	1: 1.8
SciERC	NER	1: 605.3	1: 25.5	1: 7.7
	RE	1: 913.5	1: 35.6	1: 13.6

Based on the above observations, we conclude that the two-phase paradigm allows our model to avoid suffering from grossly imbalanced data distributions.

4.4.2 Effectiveness Against Entity Length

In general, as the entity lengths increase, it becomes increasingly difficult to recognize the entities. In this section, we conduct investigations on NER performance in relation to entity lengths. We divide all entity lengths, which are restricted by the span width threshold ϵ (ϵ is set to 10), into five intervals, i.e., [1−2], [3−4], [5−6], [7−8] and [9−10]. We conduct investigations on the dev sets of the three datasets and report the results in Fig. 3. We can observe that our model consistently outperforms the baseline across all length intervals on the three datasets. Moreover, our model obtains greater F1 gains when the entity length increases. To be more precise, our model achieves the greatest improvement on ACE05 when the entity length is [7,8], and on CoNLL04 and SciERC when the entity length is [9,10], suggesting that our model is more successful in the case of long entity lengths.

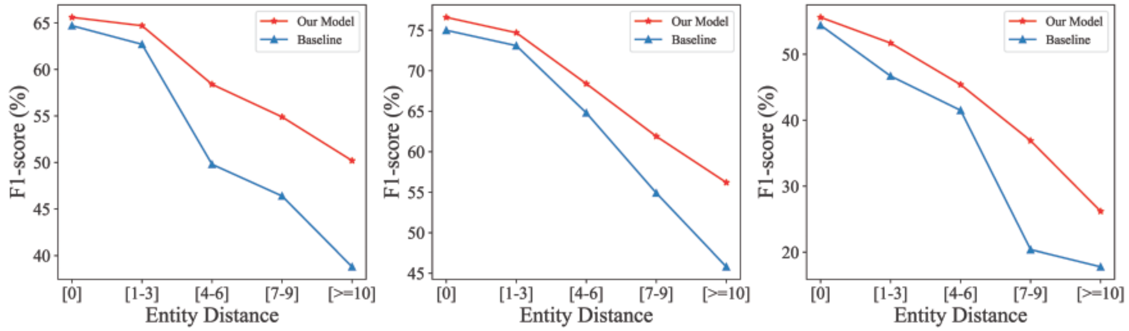


a) ACE05 b) CoNLL04 c) SciERC

Figure 3: NER performance (F1-score) comparison of our model and the baseline under various entity length intervals, which are tested on the dev sets of three datasets

4.4.3 *Effectiveness Against Entity Distance*

In general, as the distance between the two entities of a relation increases, the relation becomes more difficult to extract. In this section, we conduct investigations on RE performance in relation to entity distances. We divide all entity distances into five intervals, namely [0], [1–3], [4–6], [7–9], and [>=10]. We conduct investigations on the dev sets of the three datasets and report the investigation results in Fig. 4. The results demonstrate that our model beats the baseline across all distance intervals. Specifically, our model obtains greater improvement when the distance increases, demonstrating that our model is more effective in the case of long entity distances.



a) ACE05 a) CoNLL04 a) SciERC

Figure 4: RE performance (F1-score) comparison of our model and the baseline under various entity distance intervals, which are tested on the dev sets of three datasets

4.5 *Ablation Study*

We conduct ablation studies on the dev sets of the three datasets to analyze the effects of various model components. We report the ablation results in Tab. 7, where the "w/o Two-Phase" denotes ablating the two-phase paradigm. As a result, our approach is incapable of dealing with unbalanced data distributions. Additionally, our model cannot make use of binary global features, but retains multi-class global features. The "w/o Bi-Features" denotes ablating the binary global features, which is realized by removing $\boldsymbol{D}_{r_b}$ from $\boldsymbol{R}_{r_b}$. The "w/o Multi-Features" denotes ablating the multi-class global features, which is realized by removing $\boldsymbol{C}_r$ from $\boldsymbol{R}'_r$. The "w/o Both-Features" denotes conducting the above "w/o Bi-Features" and "w/o Multi-Features" ablations simultaneously. The "w/o gated" denotes ablating the gated mechanism. We use the concatenation manner to concatenate various semantic representations instead. The "base" denotes conducting all above ablations. After doing this, our model has the same neural architecture as SpERT.

Table 7: Ablation results on the dev sets of the three datasets. We only report the F1-scores

Model	ACE05		CoNLL04		SciERC	
	NER (F1)	RE (F1)	NER (F1)	RE (F1)	NER (F1)	RE (F1)
Our Model	88.1	64.8	88.7	74.2	71.4	53.5
w/o Two-phase	86.5(−0.6)	61.7(−3.1)	86.4(−2.3)	71.5(−2.7)	68.2(−3.2)	50.9(−2.6)
w/o Bi-features	88.3(+0.2)	63.9(−0.9)	88.2(−0.5)	73.5(−0.7)	71.1(−0.3)	51.7(−1.8)
w/o Multi-features	87.8(−0.3)	63.4(−1.4)	88.4(−0.3)	72.4(−1.8)	71.5(+0.1)	52.0(−1.5)
w/o Both-features	87.5(−0.6)	61.5(−3.3)	88.5(−0.2)	72.0(−2.2)	70.0(−0.4)	51.5(−2.0)
w/o gated	87.9(−0.2)	64.3(−0.5)	88.1(−0.6)	73.3(−0.9)	69.9(−1.5)	52.7(−0.8)
base	86.3(−1.8)	61.5(−3.3)	85.9(−2.8)	70.2(−4.0)	67.8(−3.6)	49.0(−4.5)

We have the following observations: (1) The two-phase paradigm consistently improves the model performance across the three datasets, delivering + 0.6% to + 3.2% F1-scores on NER and + 2.6% to + 3.1% F1-scores on RE, which can be attributed to the paradigm's ability to prevent our model from being harmed by grossly imbalanced data distributions. (2) Both binary and multi-class global features consistently benefit RE performance, and the multi-class features are generally more effective than the binary ones, as demonstrated on ACE05 and CoNLL04. The explanation for this could be that the multi-class features take into account fine-grained entity types. Additionally, both types of global features have a negligible effect on NER. A plausible explanation is that these features are derived from entity information and are employed in the relation extraction. (3) The combination of the two types of global features results in improved RE performance, suggesting that they have a beneficial effect on one another. (4) The proposed gated method consistently improves model performance, bringing + 0.2% to + 1.50% F1-scores on NER and + 0.5% to + 0.9% on RE, suggesting that the gated mechanism can better fuse various semantic representations.

5 Conclusion

In this paper, we propose a two-phase span-based model for the joint entity and relation extraction task, aiming to tackle the grossly imbalanced data distributions caused by the essential negative sampling. And we augment the proposed model with global features obtained by combining entity types and entity distances. Moreover, we propose a gated mechanism for effectively fusing various semantic representations. Experimental results on several datasets demonstrate that our model consistently outperforms the strongest span-based models for the joint extraction task, establishing a new standard benchmark.

Funding Statement: This research was supported by the National Key Research and Development Program [2020YFB1006302].

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] M. Eberts and A. Ulges, "Span-based joint entity and relation extraction with transformer pre-training," in *Proc. ECAI*, Santiago de Compostela, Spain, pp. 1–8, 2020.

[2] Y. Luan, L. H. He, M. Ostendorf and H. Hajishirzi, "Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction," in *Proc. EMNLP*, Brussels, Belgium, pp. 3219–3232, 2018.

[3] K. Dixit and Y. Al-Onaizan, "Span-level model for relation extraction," in *Proc. ACL*, Florence, Italy, pp. 5308–5314, 2019.

[4] Y. Luan, D. Wadden, L. H. He, A. Shah, M. Ostendorf *et al.,* "A general framework for information extraction using dynamic span graphs," in *Proc. NAACL*, Minneapolis, MN, USA, pp. 3036–3046, 2019.

[5] D. Wadden, U. Wennberg, Y. Luan and H. Hajishirzi, "Entity, relation, and event extraction with contextualized span representations," in *Proc. EMNLP*, Hong Kong, China, pp. 5783–5788, 2019.

[6] B. Ji, J. Yu, S. S. Li, J. Ma, Q. B. Wu *et al.,* "Span-based joint entity and relation extraction with attention-based span-specific and contextual semantic representations," in *Proc. COLING*, Barcelona, Spain, pp. 88–99, 2020.

[7] Z. X. Zhong and D. Q. Chen, "A frustratingly easy approach for entity and relation extraction," in *Proc. NAACL*, Online, pp. 50–61, 2021.

[8] M. Miwa and M. Bansal, "End-to-end relation extraction using LSTMS on sequences and tree structures," in *Proc. ACL*, Berlin, Germany, pp. 1105–1116, 2016.

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

[10] S. Zhao, M. H. Hu, Z. P. Cai and F. Liu, "Modeling dense cross-modal interactions for joint entity-relation extraction," in *Proc. IJCAI*, Yokohama, Japan, pp. 4032–4038, 2020.

[11] D. J. Zeng, K. Liu, S. W. Lai, G. Y. Zhou and J. Zhao, "Relation classification via convolutional deep neural network," in *Proc. COLING*, Dublin, Ireland, pp. 2335–2344, 2014.

[12] W. Ye, B. Li, R. Xie, Z. H. Sheng, L. Chen *et al.,* "Exploiting entity bio tag embeddings and multi-task learning for relation extraction with imbalanced data," in *Proc. ACL*, Florence, Italy, pp. 1351–1360, 2019.

[13] K. Lee, L. H. He, M. Lewis and L. Zettlemoyer, "End-to-end neural coreference resolution," in *Proc. EMNLP*, Copenhagen, Denmark, pp. 188–197, 2017.

[14] L. H. He, K. Lee, O. Levy and L. Zettlemoyer, "Jointly predicting predicates and arguments in neural semantic role labeling," in *Proc. ACL*, Melbourne, Australia, pp. 364–369, 2018.

[15] A. P. B. Veyseh, M. V. Nguyen, N. N. Trung, B. Min and T. H. Nguyen, "Modeling document-level context for event detection via important context selection," in *Proc. EMNLP*, Punta Cana, Dominican Republic, pp. 5403–5413, 2021.

[16] R. D. Girolamo, C. Esposito, V. Moscato and G. Sperli, "Evolutionary game theoretical on-line event detection over tweet streams," *Knowledge-Based Systems*, vol. 211, pp. 106563, 2021.

[17] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark *et al.,* "Deep contextualized word representations," in *Proc. NAACL*, New Orleans, Louisiana, pp. 2227–2237, 2018.

[18] J. Devlin, M. W. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL*, Minneapolis, Minnesota, pp. 4171–4186, 2019.

[19] Z. Z. Lan, M. D. Chen, S. Goodman, K. Gimpel, P. Sharma *et al.,* "ALBERT: A lite bert for self-supervised learning of language representations," in *Proc. ICLR*, Addis Ababa, Ethiopia, pp. 1–8, 2019.

[20] Z. Q. Geng, Y. H. Zhang and Y. M. Han, "Joint entity and relation extraction model based on rich semantics," *Neurocomputing*, vol. 429, pp. 132–140, 2021.

[21] Y. J. Wang, C. Z. Sun, Y. B. Wu, H. Zhou, L. Li *et al.,* "ENPAR: Enhancing entity and entity pair representations for joint entity relation extraction," in *Proc. ECAC*, Online, pp. 2877–2887, 2021.

[22] L. M. Hu, L. H. Zhang, C. Shi, L. Q. Nie, W. L. Guan *et al.,* "Improving distantly-supervised relation extraction with joint label embedding," in *Proc. EMNLP*, Hong Kong, China, pp. 3821–3829, 2019.

[23] A. Katiyar and C. Cardie, "Going out on a limb: Joint extraction of entity mentions and relations without dependency trees," in *Proc. ACL*, Vancouver, Canada, pp. 917–928, 2017.

[24] C. Z. Sun, Y. B. Wu, M. Lan, S. L. Sun, W. T. Wang *et al.,* "Extracting entities and relations with joint minimum risk training," in *Proc. EMNLP*, Brussels, Belgium, pp. 2256–2265, 2018.

[25] J. Wang and W. Lu, "Two are better than one: Joint entity and relation extraction with table-sequence encoders," in *Proc. EMNLP*, Online, pp. 1706–1721, 2020.

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones *et al.,* "Attention is all you need," in *Proc. NIPS*, Long Beach, CA, USA, pp. 5998–6008, 2017.

[27] K. Ding, S. S. Liu, Y. H. Zhang, H. Zhang, X. X. Zhang *et al.,* "A knowledge-enriched and span-based network for joint entity and relation extraction," Computers, *Materials & Continua*, vol. 68, no. 1, pp. 377–389, 2021.

[28] H. Y. Sun and R. Grishman, "Lexicalized dependency paths based supervised learning for relation extraction," *Computer Systems Science & Engineering*, vol. 43, no. 3, pp. 861–870, 2022.

[29] B. Ji, S. S. Li, J. Yu, J. Ma, J. T. Tang *et al.,* "Research on Chinese medical named entity recognition based on collaborative cooperation of multiple neural network models," *Journal of Biomedical Informatics*, vol. 104, pp. 103395, 2020.

[30] G. Doddington, A. Mitchell, M. Przybocki, L. Ramshaw, S. Strassel *et al.,* "The automatic content extraction (ace) program-Tasks, data, and evaluation," in *Proc. LREC*, Lisbon, Portugal, pp. 837–840, 2004.

[31] D. Roth and W. T. Yih, "A linear programming formulation for global inference in natural language tasks," in *Proc. NAACL*, Boston, Massachusetts, USA, pp. 1–8, 2004.

[32] Q. Li and H. Ji, "Incremental joint extraction of entity mentions and relations," in *Proc. ACL*, Baltimore, MD, USA, pp. 402–412, 2014.

[33] X. Y. Li, F. Yin, Z. J. Sun, X. Y. Li, A. Yuan *et al.,* "Entity-relation extraction as multi-turn question answering," in *Proc. ACL*, Florence, Italy, pp. 1340–1350, 2019.

[34] Y. J. Wang, C. Z. Sun, Y. B. Wu, H. Zhou, L. Li *et al.,* "UniRE: A unified label space for entity relation extraction," in *Proc. ACL*, Online, pp. 220–231, 2021.

[35] T. Y. S. S. Santosh, P. Chakraborty, S. Dutta, D. K. Sanyal and P. P. Das, "Joint entity and relation extraction from scientific documents: Role of linguistic information and entity types," in *Proc. EEKE*, Online, pp. 15–19, 2021.

[36] H. Y. Zhang, G. Q. Zhang and Y. Ma, "Syntax-informed self-attention network for span-based joint entity and relation extraction," *Applied Sciences*, vol. 11, no. 4, pid. 1480, pp. 1–16, 2021.

[37] X. G. Wang, D. Wang and F. P. Ji, "A span-based model for joint entity and relation extraction with relational graphs," in *Proc. IBDCloud*, Exeter, UK, pp. 513–520, 2020.

[38] Y. T. Tang, J. Yu, S. S. Li, B. Ji, Y. S. Tan *et al.,* "Span representation generation method in entity-relation joint extraction," in *Proc. ICTA*, Xi'an, China, pp. 465–476, 2021.