Synchronous Dual Network with Cross-Type Attention for Joint Entity and Relation Extraction

Hui Wu[12] and **Xiaodong Shi**[123*]
[1]Department of Artificial Intelligence, School of Informatics, Xiamen University, China
[2]National Institute for Data Science in Health and Medicine, Xiamen University, China
[3]Key Laboratory of Digital Protection and Intelligent Processing of Intangible Cultural Heritage of Fujian and Taiwan (Xiamen University), Ministry of Culture and Tourism, China
huistudent@stu.xmu.edu.cn, mandel@xmu.edu.cn

Abstract

Joint entity and relation extraction is challenging due to the complex interaction of interaction between named entity recognition and relation extraction. Although most existing works tend to jointly train these two tasks through a shared network, they fail to fully utilize the interdependence between entity types and relation types. In this paper, we design a novel synchronous dual network (SDN) with cross-type attention via *separately* and *interactively* considering the entity types and relation types. On the one hand, SDN adopts two isomorphic bi-directional type-attention LSTM to encode the entity type enhanced representations and the relation type enhanced representations, respectively. On the other hand, SDN explicitly models the interdependence between entity types and relation types via cross-type attention mechanism. In addition, we also propose a new *multi-task learning* strategy via modeling the interaction of two types of information. Experiments on NYT and WebNLG datasets verify the effectiveness of the proposed model, achieving state-of-the-art performance.

1 Introduction

Joint entity and relation extraction is a fundamental and important task in information extraction, providing necessary information for knowledge base construction (Mesquita et al., 2019), question answering (Yu et al., 2017), and dialogue systems (Xu et al., 2019), etc. This task can be decomposed into two subtasks: named entity recognition (NER) and relation extraction (RE), aiming respectively to detect text spans with specific types (entities) and semantic relations among those text spans (relations) from unstructured texts.

Early studies employ pipeline models (Zelenko et al., 2002; Zhou et al., 2005; Chan and Roth, 2011), which first extract all entities of the sentence by an entity model, and then these extracted entities are used as the inputs of a relation model to identify semantic relations between entity pairs. However, pipeline models disregard the correlation between NER and RE (Li and Ji, 2014).

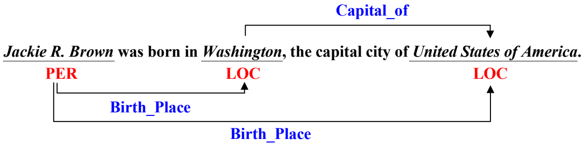

Capital_of
Jackie R. Brown was born in Washington, the capital city of United States of America.
PER
LOC
LOC
Birth_Place
Birth_Place

Figure 1: An example of joint entity and relation extraction. The italics with underline represent entities. The red and blue represent entity types and relation types, respectively.

In recent years, joint learning models have gained increasing attention. Among them, multi-task learning methods (Miwa and Bansal, 2016; Katiyar and Cardie, 2017; Fu et al., 2019; Liu et al., 2020) are popular and they utilize a shared network to learn common features, but making independent predictions for the two tasks. Later, Sun et al. (2020) propose a new multi-task learning method to dynamically learn the interactions between the two tasks. Lin et al. (2020) propose a joint neural framework OneIE to study the interaction of different feature categories by a set of global feature templates. Other methods such as novel tagging (Zheng et al., 2017; Wei et al., 2020; Wang et al., 2020) and generative models (Zeng et al., 2018, 2019; Nayak and Tou Ng, 2020; Ye et al., 2021) adopt a unified model to directly extract relational triplets. Although these methods show effectiveness for joint entity and relation extraction, they only apply a shared network or a unified model to capture the connection between NER and RE without taking into account the interdependence of **entity types** and **relation types**.

Intuitively, the relation types of relational triplets are not only relevant to the textual context and entities, but also to entity types (Peng et al., 2020).

[*] Corresponding author.

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 2769–2779
November 7–11, 2021. ©2021 Association for Computational Linguistics

Conversely, the entity types of subject and object entities are also constrained by relation types in the triplets. For example, the instance in Figure 1 contains three entities: "*Jackie R. Brown*" (PER), "*Washington*" (LOC), and "*United States of America*" (LOC). The relation type of ("*Jackie R. Brown*", "*Washington*") and ("*Jackie R. Brown*", "*United States of America*") is Birth_Place, while the relation type of ("*Washington*", "*United States of America*") is Capital_of. When the subject entity type is PER and the object entity type is LOC, the relation type between them may be Birth_Place, but never Capital_of. Conversely, when the relation type is Capital_of, the corresponding entity type does not include PER. In addition, although Sun et al. (2019) adopt graph convolutional network to handle the joint type inference problem, they fail to discuss the fine-grained correlation between entity types and relation types.

In this paper, we propose a **S**ynchronous **D**ual **N**etwork (SDN) with cross-type attention to *separately* and *interactively* capture the vector representations related to entity types and relation types. First, SDN adopts two isomorphic bi-directional type-attention LSTM to learn two different feature representations, where one is the entity type enhanced representations and the other is the relation type enhanced representations. Then the entity type information and the relation type information are introduced into the relation type enhanced representations and the entity type enhanced representations respectively to explicitly model the interaction between entity types and relation types via cross-type attention mechanism. These type-related representations are concatenated together for NER and RE via a multi-task learning strategy. The above type-attention LSTM is a general structure to select the preferred type distribution. The main idea is to inject all possible entity types simultaneously via multiple type-related cells based on the standard LSTM. In this way, our model obtains the preferred type information via training auxiliary tasks.

To summarize, the main contributions of this work are as follows[1]:

(1) We design a general type-attention LSTM structure to inject all possible type information, which can capture the preferred type features via training the corresponding auxiliary task.

(2) We propose a novel synchronous dual network with cross-type attention, which adopts cross-type attention mechanism to explicitly model the interdependence between entity types and relation types, to capture the vector representations related to entity types and relation types.

(3) Experiments on two public datasets verify the effectiveness of the multi-task learning strategy via fusing the interaction of two types of information.

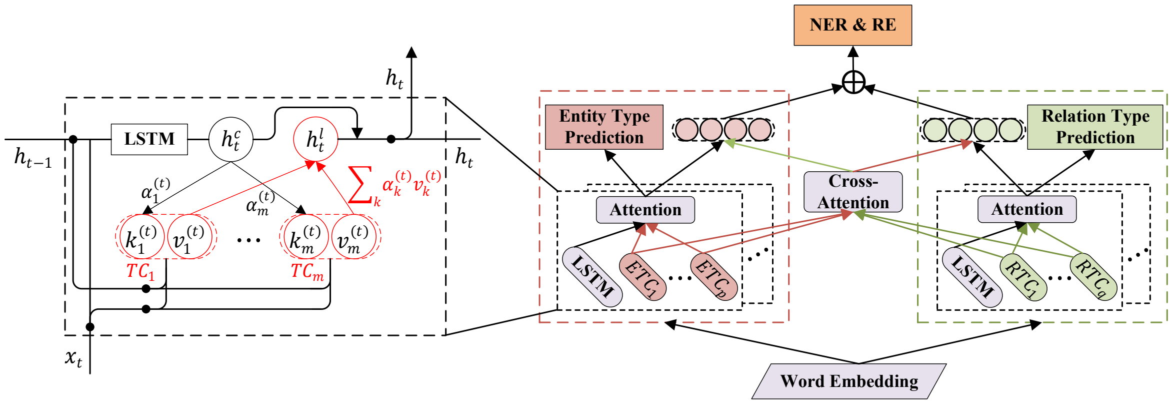
2 Related Work

Multi-Task Learning Model. Some multi-task learning models like (Miwa and Bansal, 2016; Katiyar and Cardie, 2017) learn the shared features through parameter sharing and then use them for the two subtasks of entity recognition and relation extraction. Based on this, Sun et al. (2018) enhance the interaction between the two subtasks by optimizing a global loss function. Sun et al. (2019) apply graph convolutional network to handle the interaction in type inference. These approaches essentially belong to a pipeline decoder: first extract entities and then identify the relation of the predicted entities. Later, Fu et al. (2019), Sun et al. (2020) and Liu et al. (2020) do joint learning via a shared network and simultaneously make independent predictions. Zeng et al. (2020) propose CopyMTL, a multi-task learning framework to enhance the capability of handling with multi-token entities. However, it is based on the strong assumption that the shared network is sufficient to capture the correlations between the tasks.

Tagging Model. Zheng et al. (2017) firstly convert the joint extraction task to a sequence labeling problem and propose a unified tagging scheme. Later Wei et al. (2020) and Wang et al. (2020) propose different tagging schemes. Yuan et al. (2020) propose a relation-attentive sequence labeling framework. Yu et al. (2020) adopt a novel decomposition strategy, which first recognizes head entities and then extracts corresponding tail entities and relations.

Generative Model. Zeng et al. (2018) firstly propose a sequence-to-sequence model with copy mechanism to generate relational triplets but fail to generate multi-word entities. Subsequently, Zeng et al. (2019), Nayak and Tou Ng (2020) and Sui et al. (2020) adopt different encoder-decoder architectures to generate relational triplets. Ye et al. (2021) propose contrastive triple extraction with a generative transformer.

[1] Our code will be released on Github at https://github.com/HuiStudent/Cross-type_SDN.



(a) Type-Attention LSTM unit

(b) Synchronous Dual Network with Cross-Type Attention

Figure 2: Overall structures. The red in (a) represents the calculation of type-attention mechanism and "TC_m" represents the m-th type-related cell. "ETC_p" and "RTC_q" in (b) represent the p-th entity type cell and q-th relation type cell, respectively. $\bigoplus$ denotes concatenation operation.

Compared with these works, we propose a new multi-task learning strategy to explicitly model the interdependence between entity types and relation types within one relation to synchronize their information and make them mutually beneficial.

3 Type-Attention LSTM

In this section, we will introduce the general framework of **T**ype **A**ttention **LSTM** (TA-LSTM). As shown in Figure 2 (a), each TA-LSTM unit is composed of a LSTM unit and a type-attention unit. For each token w_t, the LSTM unit is used to obtain the contextual representation $\mathbf{h}_t^c$, while the type-attention unit uses the scaled dot-product attention (Vaswani et al., 2017) to obtain the type representation $\mathbf{h}_t^l$ by integrating m type-related cells for controlling type information flow.

We adopt the standard LSTM (Graves and Schmidhuber, 2005) to encode the contextual representation for each token. At each time step $t(t \in [1, ..., n])$, the current hidden state $\mathbf{h}_t^c$ based on a memory cell $\mathbf{c}_t$ is calculated as follows:

$$\begin{bmatrix} \mathbf{i}_t \\ \mathbf{o}_t \\ \mathbf{f}_t \\ \widetilde{\mathbf{c}}_t \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{bmatrix} (\mathbf{W}[\mathbf{h}_{t-1}; \mathbf{x}_t] + \mathbf{b}) \qquad (1)$$
$$\mathbf{c}_t = \mathbf{i}_t \odot \widetilde{\mathbf{c}}_t + (\mathbf{1} - \mathbf{i}_t) \odot \mathbf{c}_{t-1}$$
$$\mathbf{h}_t^c = \mathbf{o}_t \odot \tanh(\mathbf{c}_t),$$

where $[\mathbf{W}; \mathbf{b}]$ are trainable parameters. $\mathbf{x}_t \in \mathbb{R}^{d_w}$ is the word embedding of each token w_t. σ represents the sigmoid activation function. $\mathbf{i}_t$, $\mathbf{o}_t$, and $\mathbf{f}_t$ represent the input gate, output gate, and forget gate, respectively. The hidden state $\mathbf{h}_t^c \in \mathbb{R}^{d_e}$ represents the vector representation with context information.

Type-Attention Mechanism. Given $\mathbf{x}_t$ and $\mathbf{h}_{t-1}$, the key-value pair of the k-th ($k \in [1, \ldots, m]$) type-related cell is computed as:

$$\begin{bmatrix} \mathbf{k}_k^{(t)} \\ \mathbf{v}_k^{(t)} \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \end{bmatrix} (\mathbf{W}_k[\mathbf{h}_{t-1}; \mathbf{x}_t] + \mathbf{b}_k), \qquad (2)$$

where $[\mathbf{W}_k; \mathbf{b}_k]$ represent the trainable parameters specific to the k-th type-related cell. σ represents the sigmoid activation function. $\mathbf{k}_k^{(t)} \in \mathbb{R}^{d_e}$ and $\mathbf{v}_k^{(t)} \in \mathbb{R}^{d_e}$ represent the key and value of the k-th type-related cell, respectively.

The above operations are repeated for m type-related cells. At the time step t, we finally acquire a set of key-value pairs $\mathbf{K}^{(t)} = [\mathbf{k}_1^{(t)}, \ldots, \mathbf{k}_m^{(t)}]$ and $\mathbf{V}^{(t)} = [\mathbf{v}_1^{(t)}, \ldots, \mathbf{v}_m^{(t)}]$.

We regard the contextual representation $\mathbf{h}_t^c$ as the query. The scaled dot-product attention first computes the dot products of the query with all corresponding keys, divide each by $\sqrt{d_e}$, and apply a softmax function to obtain the weight $\boldsymbol{\alpha}^{(t)}$ on the values. Then the type representation $\mathbf{h}_t^l$ is computed as a weighted sum of the values:

$$\mathbf{h}_t^l = \text{attention}(\mathbf{h}_t^c, \mathbf{K}^{(t)}, \mathbf{V}^{(t)}) = \boldsymbol{\alpha}^{(t)}\mathbf{V}^{(t)}$$
$$\boldsymbol{\alpha}^{(t)} = \text{softmax}\left(\frac{\mathbf{h}_t^c \mathbf{K}^{(t)\top}}{\sqrt{d_e}}\right) \qquad (3)$$

Finally, the hidden state $\mathbf{h}_t \in \mathbb{R}^{d_e}$ of each TA-LSTM unit is computed as:

$$\mathbf{h}_t = \mathbf{h}_t^c + \mathbf{h}_t^l \qquad (4)$$

4 Synchronous Dual Network with Cross-Type Attention

Given a sentence $\mathbf{s} = [w_1, \ldots, w_n]$ of n words, joint entity and relation extraction task aims to identify a collection of relation triplets $\mathcal{T} = \{(\mathbf{e}_i, r, \mathbf{e}_j)|\mathbf{e}_i, \mathbf{e}_j \in \mathcal{E}, r \in \mathcal{R}\}$, where $\mathbf{e}_i$, $\mathbf{e}_j$, and r represent the subject entity, the object entity, and their semantic relation, respectively. Note that the subject and object entities belong to the set of entities $\mathcal{E} = \{\mathbf{e}_i\}_{i=1}^{P}$ existing in the sentence and the relation is selected from a pre-defined set $\mathcal{R} = \{\mathcal{R}_1, \ldots, \mathcal{R}_M\}$ with M valid relation types.

In this section, we will describe how to build a synchronous dual network on top of the proposed TA-LSTM and how to interact between entity types and relation types via cross-type attention mechanism for joint entity and relation extraction. As shown in Figure 2 (b), synchronous dual network adopts two bi-directional TA-LSTM to encode the vector representations with corresponding type information from the perspective of entity types and relation types by synchronous dual learning, respectively. Then synchronous dual learning adopts the cross-type attention mechanism to model the interaction between entity types and relation types through introducing entity type information into relation type learning or introducing relation type information into entity type learning. Finally, the vector representations with type information are concatenated together for joint learning.

4.1 Synchronous Dual Learning

We design entity type learning and relation type learning to obtain the entity type enhanced representation $\mathbf{h}_t^e$ and the relation type enhanced representation $\mathbf{h}_t^r$, respectively.

Entity Type Learning. To encode the entity type information, we first define p entity type cells (ETCs) according to the number of entity types. Following Jia and Zhang (2020), we regard each entity type as a label, such as PER, LOC, ORG, and so on. Entity type learning mainly aims at extracting entity type knowledge from a set of labeled training data by training entity type cells.

From Equation (1) and (2), we can acquire the contextual representation of $\bar{\mathbf{h}}_t^c = [(\overrightarrow{\mathbf{h}}_t^c + \overleftarrow{\mathbf{h}}_t^c)/2]$ and a set of key-value pairs $\bar{\mathbf{K}}^{(t)} = [\bar{\mathbf{k}}_1^{(t)}, \ldots, \bar{\mathbf{k}}_p^{(t)}]$ and $\bar{\mathbf{V}}^{(t)} = [\bar{\mathbf{v}}_1^{(t)}, \ldots, \bar{\mathbf{v}}_p^{(t)}]$ related to entity types, where $\bar{\mathbf{k}}_l^{(t)} = [(\overrightarrow{\mathbf{k}}_l^{(t)} + \overleftarrow{\mathbf{k}}_l^{(t)})/2]$ and $\bar{\mathbf{v}}_l^{(t)} = [(\overrightarrow{\mathbf{v}}_l^{(t)} + \overleftarrow{\mathbf{v}}_l^{(t)})/2]$, $(l \in [1, \ldots, p])$. The contextual representation and the type representation related to entity types are computed for the final vector representation $\mathbf{h}_t^e = [\overrightarrow{\mathbf{h}}_t^e \oplus \overleftarrow{\mathbf{h}}_t^e]$ in Equation (4). Finally the sequence representations associated to entity types are $\mathbf{H}^{(e)} = [\mathbf{h}_1^e, \ldots, \mathbf{h}_n^e]$.

To explicitly highlight entity type knowledge, we design **entity type prediction** as an auxiliary task in Figure 3 (a). Given a sentence $\mathbf{s}$ and its corresponding entity types $\mathbf{T}_e = [T_1^e, \ldots, T_n^e]$ ($T_t^e \in$ [O, PER, LOC, ORG, ...] and $t \in [1, \ldots, n]$), we regard entity type prediction as the sequence tagging problem by converting the attention scores to the aligned entity type distribution for w_t:

$$p(T_l^e|w_t) = \text{softmax}(\frac{\bar{\mathbf{h}}_t^c \bar{\mathbf{k}}_l^{(t)\top}}{\sqrt{d_e}}) \qquad (5)$$

And the negative log-likelihood loss is used for training on the sentence $\mathbf{s}$:

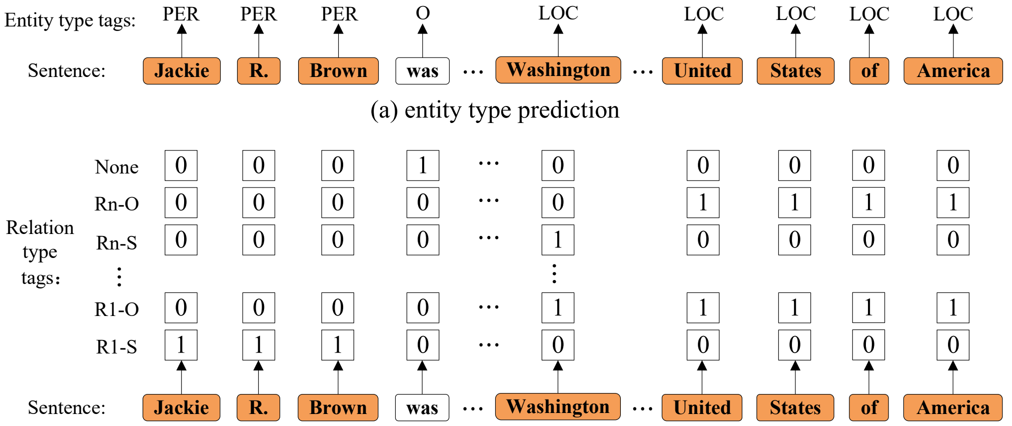
$$\mathcal{L}_{ET} = -\sum_{t=1}^{n} \log(p(T_t^e|w_t)) \qquad (6)$$

Relation Type Learning. Similar to entity type learning, q kinds of new relation types are designed to correspond to entity types by considering M valid relation types of the pre-defined set $\mathcal{R}$ and the triplet composition. As shown in Figure 3 (b), these new relation types are composed of relation labels and subject or object labels, namely "R$_1$-S", "R$_1$-O", ..., "R$_M$-S", "R$_M$-O" and "None" ("None" represents the invalid relation type). Obviously, the number of new relation types $q = 2 \times M + 1$.

These new relation types are mainly used to learn the distribution of the subject or object type affected by valid relation types. Thus, we design relation-related TA-LSTM as the other model of synchronous dual network to extract new relation type knowledge.

At the time step t, we can acquire the contextual representation of $\hat{\mathbf{h}}_t^c = [(\overrightarrow{\mathbf{h}}_t^c + \overleftarrow{\mathbf{h}}_t^c)/2]$ and a set of key-value pairs $\hat{\mathbf{K}}^{(t)} = [\hat{\mathbf{k}}_1^{(t)}, \ldots, \hat{\mathbf{k}}_q^{(t)}]$ and $\hat{\mathbf{V}}^{(t)} = [\hat{\mathbf{v}}_1^{(t)}, \ldots, \hat{\mathbf{v}}_q^{(t)}]$ related to new relation types, where $\hat{\mathbf{k}}_l^{(t)} = [(\overrightarrow{\mathbf{k}}_l^{(t)} + \overleftarrow{\mathbf{k}}_l^{(t)})/2]$ and $\hat{\mathbf{v}}_l^{(t)} = [(\overrightarrow{\mathbf{v}}_l^{(t)} + \overleftarrow{\mathbf{v}}_l^{(t)})/2]$, $(l \in [1, \ldots, q])$ in Equation (1-2). Finally, the sequence representations associated to new relation types are $\mathbf{H}^{(r)} = [\mathbf{h}_1^r, \ldots, \mathbf{h}_n^r]$, where the vector representation $\mathbf{h}_t^r = [\overrightarrow{\mathbf{h}}_t^r \oplus \overleftarrow{\mathbf{h}}_t^r]$ is computed by Equation (4).

Different from entity type prediction, the second auxiliary task **relation type prediction** is a multi-label classification problem because the same entity



(a) entity type prediction

(b) relation type prediction

Figure 3: Entity type prediction & relation type prediction. The whole sentence in (a) and (b) comes from Figure 1. We omit some words for simplicity. Entity type prediction and relation type prediction are regarded as different sequence tagging tasks to learn entity types and relation types, respectively. “R1” and “Rn” represent relation labels; “R1-S” and “R1-O” represent the labels of the subject and object entities in the relation triplet (S, R1, O), respectively.

may exist in multiple relational triplets (namely overlapping triplets). We adopt multiple identical binary classifiers to detect different relation types by assigning each token a binary tag $(0/1)$ that indicates whether the current token corresponds to a new relation type. So the aligned relation type distribution for w_t is computed as:

$$p(T_l^r|w_t) = \text{sigmoid}\left(\frac{\hat{\mathbf{h}}_t^c \hat{\mathbf{k}}_l^{(t)\top}}{\sqrt{d_e}}\right) \qquad (7)$$

where T_l^r belongs to q new relation type labels, such as “R$_1$-S”, “R$_1$-O” and so on.

The binary cross-entropy loss is used for training on the sentence $\mathbf{s}$:

$$\mathcal{L}_{RT} = -\sum_{t=1}^{n}\sum_{r=1}^{q}\{\log(p(T_t^r|w_t))^{\mathbb{I}\{\hat{T}_t^r=1\}} + \log(1-p(T_t^r|w_t))^{\mathbb{I}\{\hat{T}_t^r=0\}}\} \qquad (8)$$

where $\hat{T}_t^r$ represents the gold relation type label.

4.2 Cross-Type Attention Mechanism

Since entity types and relation types are interdependent in the joint entity and relation extraction task, it is important to synchronize their information and make them mutually beneficial. So we propose a novel cross-type attention mechanism to model the interaction between entity types and relation types.

Given the entity type enhanced representation $\mathbf{h}_t^e$ and a set of key-value pairs $\hat{\mathbf{K}}^{(t)}$ and $\hat{\mathbf{V}}^{(t)}$ related to new relation types, the relation-entity representation $\mathbf{c}_t^e$ is computed by Equation (3) via introducing relation type information into entity type learning.

Similarly, given $\mathbf{h}_t^r$ and the entity type key-value pairs $\bar{\mathbf{K}}^{(t)}$ and $\bar{\mathbf{V}}^{(t)}$, we can obtain the entity-relation representation $\mathbf{c}_t^r$. Finally, we obtain the new entity type enhanced representation $\tilde{\mathbf{h}}_t^e$ and new relation type enhanced representation $\tilde{\mathbf{h}}_t^r$ to model the interdependency between entity types and relation types by adding $\mathbf{c}_t^e$ and $\mathbf{c}_t^r$ into $\mathbf{h}_t^e$ and $\mathbf{h}_t^r$, respectively.

4.3 Joint Entity and Relation Extraction

We first concatenate the vector representations $\tilde{\mathbf{h}}_t^e$ and $\tilde{\mathbf{h}}_t^r$ to obtain the overall representation $\tilde{\mathbf{h}}_t$:

$$\tilde{\mathbf{h}}_t = \tilde{\mathbf{h}}_t^e \oplus \tilde{\mathbf{h}}_t^r \qquad (9)$$

Then the overall sequence representation $\tilde{\mathbf{H}} = [\tilde{\mathbf{h}}_1, \ldots, \tilde{\mathbf{h}}_n]$ is used for NER and RE.

Named Entity Recognition. NER is a typical sequence labeling task. Here we use the BIESO tagging as the tagging scheme to recognize the entity boundary accurately. Given a sentence $\mathbf{s}$, the probability distribution y_t of a word w_t over these five labels is calculated based on the overall representation $\tilde{\mathbf{h}}_t \in \tilde{\mathbf{H}}$ as follows:

$$y_t = \text{softmax}(\mathbf{W}_e\tilde{\mathbf{h}}_t + \mathbf{b}_e) \qquad (10)$$

where $[\mathbf{W}_e; \mathbf{b}_e]$ are trainable parameters.

The negative log-likelihood loss is used for training on the sentence $\mathbf{s}$:

$$\mathcal{L}_E = -\sum_{t=1}^{n}\log(y_t) \qquad (11)$$

Relation Extraction. Following Sun et al. (2020), we classify all relations between pairs of words in the sentence based on the overall representation $\tilde{\mathbf{H}}$. Similar to relation type prediction, the relation extraction task belongs to a multilabel classification problem. We design multiple identical binary classifiers to detect different relations.

The probability distribution $y^{r'}_{i,j}$ of the word pair (w_i, w_j) belonging to the relation $r' \in \mathcal{R}$ is computed as follows:

$$\begin{aligned} \mathbf{m} &= \phi(\mathbf{W}_m(\tilde{\mathbf{h}}_i \oplus \tilde{\mathbf{h}}_j) + \mathbf{b}_m) \\ y^{r'}_{i,j} &= \mathrm{sigmoid}(\mathbf{W}_{r'}\mathbf{m} + \mathbf{b}_{r'}) \end{aligned} \tag{12}$$

where $[\mathbf{W}_m; \mathbf{b}_m; \mathbf{W}_{r'}; \mathbf{b}_{r'}]$ are trainable parameters. ϕ represents the ReLU activation function.

The binary cross-entropy loss is used for training on the sentence $\mathbf{s}$:

$$\begin{aligned} \mathcal{L}_R = -\sum_{r'=1}^{M}\sum_{i,j=1}^{n} \{ & \log(y^{r'}_{i,j})^{\mathbb{I}\{\hat{y}^{r'}_{i,j}=1\}} \\ & + \log(1 - y^{r'}_{i,j})^{\mathbb{I}\{\hat{y}^{r'}_{i,j}=0\}} \} \end{aligned} \tag{13}$$

where $\hat{y}^{r'}_{i,j}$ represents the gold relation label.

4.4 Training

Given a training instance $\mathbf{s}$, the training objective of joint entity and relation extraction is as follows:

$$\mathcal{L} = \lambda^{t1}\mathcal{L}_{ET} + \lambda^{t2}\mathcal{L}_{RT} + \lambda^{e}\mathcal{L}_{E} + \lambda^{r}\mathcal{L}_{R} + \frac{\lambda}{2}\|\Theta\|^2 \tag{14}$$

where λ^{t1}, λ^{t2}, λ^{e}, and λ^{r} represent the different task weights. λ is the L_2 regularization parameters and Θ represents the parameters set.

4.5 Inference

Following Fu et al. (2019), we adopts the average prediction method to infer whether the extracted triplet is right. Concretely, we can obtain the entity set $\mathcal{E}$ by named entity recognition. Then, given the subject entity $\mathbf{e}_i = [w_{\xi_i}, \ldots, w_{\zeta_i}]$ and the object entity $\mathbf{e}_j = [w_{\xi_j}, \ldots, w_{\zeta_j}]$, the probability p_r that they belong to the r-th relation type can be calculated as follows:

$$p_r = \frac{1}{|\mathbf{e}_i|}\frac{1}{|\mathbf{e}_j|}\sum_{f=\xi_i}^{\zeta_i}\sum_{s=\xi_j}^{\zeta_j} y^{r}_{f,s} \tag{15}$$

where $|\mathbf{e}_i|$ and $|\mathbf{e}_j|$ represent the length of $\mathbf{e}_i$ and $\mathbf{e}_j$, respectively. The triplet $(\mathbf{e}_i, r, \mathbf{e}_j)$ is extracted only if $p_r > \theta$, where θ is a free threshold parameter.

Class	NYT			WebNLG		
	Train	Dev	Test	Train	Dev	Test
#ALL	56195	5000	5000	5019	500	703
#Entity		3			15	
#Relation		24			246	

Table 1: Statistics of datasets. #ALL, #Entity, and #Relation represent the total number of sentences, entity types, and relation types, respectively.

5 Experiments

In this section, we empirically verify the effectiveness of our proposed SDN on two public datasets. In addition, we also investigate how different components in the model impact the performance of joint entity and relation extraction with different settings.

5.1 Experimental Settings

Datasets. We conduct experiments to evaluate SDN on two public datasets NYT (Riedel et al., 2010) and WebNLG (Gardent et al., 2017). The NYT dataset is produced by the distant supervision method which automatically aligns Freebase with New York Times news articles. It includes 3 entity types (e.g., “PER”, “LOC”, and “ORG”) and 24 valid relation types. The WebNLG dataset is originally created for the natural language generation task. Given a group of triplets, annotators are asked to write a sentence which contains the information of all triplets in this group. We directly use the dataset preprocessed by Zeng et al. (2018), which includes 15 entity types and 246 valid relation types. The statistics of NYT and WebNLG are described in Table 1.

Hyperparameters. We initialize the word embeddings with Glove 300-dimension vectors (Pennington et al., 2014). The dimensions of hidden states for TA-LSTM, entity extraction module, and relation extraction module are set to 100, 200, 400, respectively. During training, we use the Adam optimizer with the initial learning rates of $1e^{-3}$ on NYT and $5e^{-4}$ on WebNLG, a maximum epoch number of 100 and the batch size of 30. To avoid overfitting, we apply $\mathrm{Dropout}$ with a rate of 0.3. In $\mathrm{SDN}_{\text{CROSS-TA-LSTM}} + \mathrm{BERT}$ and $\mathrm{SDN}_{\text{TA-LSTM}} + \mathrm{BERT}$, we use $\mathrm{bert\text{-}base\text{-}cased}$[2] to initialize BERT (Devlin et al., 2018) and the initial learning rates is $1e^{-5}$ for fine-tuning BERT.

[2] https://storage.googleapis.com/bert_models/2018_10_18/cased_L-12_H-768_A-12.zip

Methods		NYT			WebNLG		
		P	R	F$_1$	P	R	F$_1$
Multi-task	MRT (Sun et al., 2018)	65.2	40.6	50.0	-	-	-
	RIN (Sun et al., 2020)	83.9	85.5	84.7	77.3	76.8	77.0
	CopyMTL-One (Zeng et al., 2020)	72.7	69.2	70.9	57.8	60.1	58.9
	CopyMTL-Mul (Zeng et al., 2020)	75.7	68.7	72.0	58.0	54.9	56.4
Tagging	PA-LSTM-CRF (Dai et al., 2019)	49.4	59.1	53.8	-	-	-
	RSAN (Yuan et al., 2020)	85.7	83.6	84.6	80.5	83.8	82.1
	TPLinker(LSTM) (Wang et al., 2020)	86.0	82.0	84.0	$\underline{91.9}$	81.6	86.4
	TPLinker(BERT)† (Wang et al., 2020)	91.4	**92.6**	92.0	88.9	84.5	86.7
Generative	OneDecoder (Zeng et al., 2018)	59.4	53.1	56.0	32.2	28.9	30.5
	MultiDecoder (Zeng et al., 2018)	61.0	56.6	58.7	37.7	36.4	37.1
	PNDec (Nayak and Tou Ng, 2020)	89.3	78.8	83.8	-	-	-
	CGT(BERT)† (Ye et al., 2021)	**94.7**	84.2	89.1	**92.9**	75.6	83.4
Ours	SDN$_{\text{LSTM}}$	86.5	81.7	84.1	85.1	82.5	83.7
	SDN$_{\text{TA-LSTM}}$	89.8	84.0	86.8	88.2	85.9	87.0
	SDN$_{\text{Cross-TA-LSTM}}$	$\underline{90.1}$	$\underline{86.2}$	$\underline{88.1}$	90.8	$\underline{88.8}$	$\underline{89.8}$
	SDN$_{\text{TA-LSTM}}$ + BERT†	93.7	89.1	91.3	91.8	88.8	90.3
	SDN$_{\text{Cross-TA-LSTM}}$ + BERT†	94.2	91.5	**92.8**	92.7	**89.6**	**91.1**

Table 2: Results of different methods on NYT and WebNLG datasets. The reported results for the above baselines are directly copied from the original published literature. † indicates the method leveraging the pre-trained language model BERT. The bold represents the best results. The underlined represents the best results without BERT.

All experiments are conducted with an NVIDIA GeForce RTX 2080 Ti.

Baselines and Evaluation Metrics. We compare our method against three groups of state-of-the-art methods of joint learning:

i) Multi-task baselines. MRT (Sun et al., 2018) applies a minimum risk training method to highlight connections between an entity model and a relation model. RIN (Sun et al., 2020) uses a recurrent interaction network to dynamically learn. SMHSA (Liu et al., 2020) uses an attention-based joint model to identify the overlapping triplets. CopyMTL-One and CopyMTL-Mul (Zeng et al., 2020) utilize a multi-task framework to extract multi-token entities.

ii) Tagging baselines. PA-LSTM-CRF (Dai et al., 2019) uses a position-attention mechanism to model n tag sequences. RSAN (Yuan et al., 2020) utilizes a relation-attentive sequence labeling framework. TPLinker (Wang et al., 2020) designs a handshaking tagging scheme for joint entity and relation extraction.

iii) Generative baselines. OneDecoder and MultiDecoder (Zeng et al., 2018) utilize a Seq2Seq model to generate relational triplets. PNDec (Nayak and Tou Ng, 2020) uses pointer networks with an encoder-decoder model. CGT (Ye et al., 2021) adopts a framework of contrastive triple extraction with a generative transformer.

We adopt Precision(P), Recall(R), and standard Micro-F1 (F$_1$) to evaluate the performance. A predicted triplet is regarded as correct only if the relation type and the two corresponding entities are all the same as the golden standard annotation[3]. We report the corresponding results of the test set when the development set achieves the best results.

5.2 Main Results

As shown in Table 2, SDN$_{\text{Cross-TA-LSTM}}$ is significantly superior to these multi-task baselines (over at least 6.2% P, 0.7% R, 3.4% F$_1$ on NYT and 13.5% P, 12.0% R, 12.8% F$_1$ on WebNLG). It shows that explicitly modeling the interdependence of entity types and relation types related to entities in the NER and RE tasks can obtain more useful representations for joint entity and relation extraction.

Without considering the pre-training language model BERT, SDN$_{\text{Cross-TA-LSTM}}$ achieves better performances than other tagging baselines and generative baselines (over at least 0.8% P, 2.6% R, 3.5% F$_1$ on NYT and 5.0% R, 3.4% F$_1$ on WebNLG). This indicates that our model can predict as many accurate triples as possible while ensuring a higher precision.

Recently, pre-trained language models such as

[3] For a fair comparison with previous recent works, we select only the baselines with the exact match criteria.

Methods	NER		RE	
	F_1	ΔF_1	F_1	ΔF_1
Ours	**94.6**	-	**92.8**	-
$-\mathcal{L}_{ET}$	93.1	-1.5	91.4	-1.4
$-\mathcal{L}_{RT}$	93.7	-0.9	91.8	-1.0
$-\mathcal{L}_{ET}-\mathcal{L}_{RT}$	91.2	-3.4	88.2	-4.6

Table 3: Ablation experiments on the NYT dataset.

BERT (Devlin et al., 2018) achieve state-of-the-art results on joint entity and relation extraction. To make a fair and direct comparison with our method, we leverage the output of BERT as contextualized word embeddings. $\text{SDN}_{\text{Cross-TA-LSTM}}$ + BERT achieves better results than TPLinker(BERT) (over 2.8% P, 0.8% F_1 on NYT and 3.8% P, 5.1% R, 4.4% F_1 on WebNLG). $\text{SDN}_{\text{Cross-TA-LSTM}}$ + BERT outperforms CGT(BERT) over 7.3% R, 3.7% F_1 on NYT and over 14.0% R, 7.7% F_1 on WebNLG, and $\text{SDN}_{\text{Cross-TA-LSTM}}$ + BERT is comparable to CGT(BERT) in Precision (94.2% P v.s. 94.7% P on NYT and 92.7% P v.s. 92.9% P on WebNLG).

Generally speaking, generative triple extraction requires a huge amount of manual annotation or further constraint to improve the precision. But our model focuses on how to extract more relation triples, which indicates the importance of the entity type and relation type information for joint entity and relation extraction. In fact, injecting the effective entity and relation type information via the interaction of entity types and relation types is promising for joint entity and relation extraction.

To more explicitly illustrate the effectiveness of our method, we make comparisons with some deformed models: (i) SDN_{LSTM}, which only uses bi-directional LSTM as encoders without any type knowledge and the cross-type interaction; (ii) $\text{SDN}_{\text{TA-LSTM}}$, which uses two bi-directional TA-LSTM to encode different type information and only introduce the knowledge of entity and relation type without considering cross-type interaction; (iii) $\text{SDN}_{\text{Cross-TA-LSTM}}$, which not only introduces the type knowledge but also models the interaction between entity and relation types; (iv) $\text{SDN}_{\text{TA-LSTM}}$ + BERT, which adds BERT to (ii) $\text{SDN}_{\text{TA-LSTM}}$; (v) $\text{SDN}_{\text{Cross-TA-LSTM}}$+BERT, which adds BERT to (iii) $\text{SDN}_{\text{Cross-TA-LSTM}}$. $\text{SDN}_{\text{TA-LSTM}}$ achieves better results of 2.7% F_1 on NYT and 3.3% F_1 on WebNLG compared with SDN_{LSTM}, indicating the effectiveness of TA-LSTM for capturing more type informations. $\text{SDN}_{\text{Cross-TA-LSTM}}$ outperforms $\text{SDN}_{\text{TA-LSTM}}$ over 1.3% F_1 on NYT and over 2.8% F_1 on WebNLG, which shows that explicitly modeling the interaction between entity and relation types within one relation can further improve performance. In addition, $\text{SDN}_{\text{Cross-TA-LSTM}}$ + BERT and is significantly superior to $\text{SDN}_{\text{TA-LSTM}}$ + BERT (over 1.5% F_1 on NYT and 0.8% F_1 on WebNLG), indicating the effectiveness of cross-type attention mechanism.

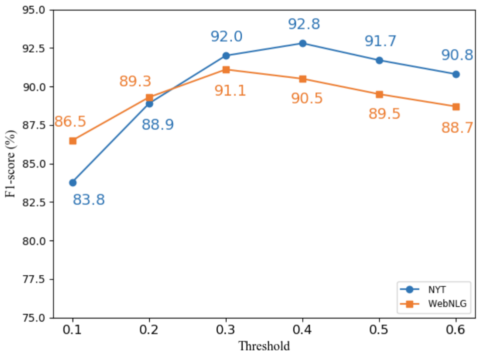


Figure 4: Analysis of the inference threshold θ

5.3 Ablation Experiments

We conduct ablation experiments of auxiliary tasks on NYT. The results are listed in Table 3.

When we only ablate $\mathcal{L}_{ET}$, the results on the two subtasks suffer significant declines (-1.5% F_1 on NER and -1.4% F_1 on RE, respectively). When we only ablate $\mathcal{L}_{RT}$, the results on NER and RE also suffer significant declines (over -0.9% F_1 on both subtasks). When we both ablate $\mathcal{L}_{ET}$ and $\mathcal{L}_{RT}$, our model achieves similar results (-3.4% F_1 on NER and -4.6% F_1 on RE, respectively). On the one hand, it demonstrates that two auxiliary tasks of entity type prediction and relation type prediction can highlight useful type information as well as decrease the noise. On the other hand, it indicates that our model depends heavily on both auxiliary tasks to capture the right type of knowledge.

5.4 Analysis of Inference Threshold

We conduct analysis experiments to explore the value of threshold θ of the inference methods. Figure 4 illustrates how performance varies on different inference threshold θ of the NYT and WebNLG datasets. It can be seen that the threshold inference method effectively adjusts the performance

Sentence	Golden Result	Result
"This is rare, and it's messy," said Gary Chaison, professor of industrial relations at Clark University in Worcester, Mass..	(Gary Chaison, /business/person/company, Clark University) (Gary Chaison, /people/person/place_lived, Worcester) (Worcester, /location/location/contains, Clark University)	**RIN**: (Gary Chaison, /business/person/company, Clark University) (Worcester, /location/location/contains, Clark University) **SDN$_{\text{TA-LSTM}}$**: (Gary Chaison, /business/person/company, Clark University) (Gary Chaison, /people/person/place_lived, Worcester) (Worcester, /location/location/contains, Clark University)
An article yesterday about the Darfur crisis in Sudan misstated a Security Council response to a report to the Council by a United Nations envoy, Ahmedou Ould -Abdallah.	(Darfur, /location/administrative_division/country, Sudan) (Sudan, /location/country/administrative_divisions, Darfur) (Sudan, /location/location/contains, Darfur)	**RIN**: (Darfur, /location/administrative_division/country, Sudan) (Sudan, /location/administrative_division/country, Darfur) ✗ (Sudan, /location/location/contains, Darfur) **SDN$_{\text{TA-LSTM}}$**: (Darfur, /location/administrative_division/country, Sudan) (Sudan, /location/country/administrative_divisions, Darfur) (Sudan, /location/location/contains, Darfur)

Table 4: Examples from the NYT test set. Red, blue, and green represent entities whose types are PER, LOC, and ORG, respectively. ✗ denotes incorrent relational triplets.

of SDN with different choices of θ. With the increase of θ from 0.1 to 0.6, the F_1 scores on NYT and WebNLG show a consistent trend of first increasing and then decreasing. When the θ values of NYT and WebNLG are set to 0.4 and 0.3 respectively, the average inference method achieves the best performance on both datasets. Through detailed analysis of NYT and WebNLG, we find that this gap is due to the difference of the entity length. More precisely, the average entity length and max entity length on the NYT test set are 1.4 and 8, respectively. while the average entity length and max entity length on the WebNLG test set are 2.2 and 15, respectively. Longer entities are harder to identify entities and relations in the NER and RE tasks.

5.5 Case Study

As shown in Table 4, we present two examples from the NYT test set as illustrations. The relation type of "(Gary Chaison (PER) , Worcester (LOC))" is "/people/person/place_lived". RIN only identifies "(Gary Chaison, /business/person/company, Clark University)" and "(Worcester, /location/location/contains, Clark University)", but fails to identify "(Gary Chaison, /people/person/place_lived, Worcester)". In addition, RIN does not recognize "(Sudan, /location/country/administrative_divisions, Darfur)" correctly. Because RIN dynamically learns the interaction of the two subtasks without considering any type of information, and some details may be lost to some extent. In contrast, our method correctly extracts all relation triplets, which shows that explicitly modeling the interaction between entity types and relation types can synchronize their information and make them mutually beneficial for joint entity and relation extraction.

6 Conclusion

In this paper, we propose a **S**ynchronous **D**ual **N**etwork (SDN) with cross-type attention for joint entity and relation extraction. Firstly, we use two isomorphic bi-directional type-attention LSTM as encoders to learn the entity type enhanced representations and the relation type enhanced representations from two different perspectives. Then the entity type information (or the relation type information) is introduced into the relation type enhanced representations (or the entity type enhanced representations) to explicitly model the interaction between entity types and relation types via cross-type attention mechanism. In addition, the proposed type-attention LSTM is a general structure to obtain the preferred type distribution. Experiments on two public datasets verify the effectiveness of the proposed model, achieving state-of-the-art performance.

Acknowledgements

We thank the anonymous reviewers for their helpful comments and suggestions. This work is supported the Major Scientific Research Project of the State Language Commission in the 13th Five-Year Plan (Grant nos. WT135-38), the Key Support Project of NSFC-Liaoning Joint Foundation(Grant no. U1908216), and the Project of Technological Innovation 2030 "New Generation Artificial Intelligence" (Grant no. 2020AAA0107904).

References

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.

Dai Dai, Xinyan Xiao, Yajuan Lyu, Shan Dou, Qiaoqiao She, and Wang Haifeng. 2019. Joint extraction of entities and overlapping relations using position-attentive sequence labeling. In *Proceedings of the 33th AAAI Conference on Arificial Intelligence*, pages 6300–6308.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating training corpora for NLG micro-planners. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 179–188.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural networks*, 18(5-6):602–610.

Chen Jia and Yue Zhang. 2020. Multi-cell compositional LSTM for NER domain adaptation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5906–5917.

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.

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.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7999–8009.

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 29th International Joint Conference on Arificial Intelligence*, pages 3787–3793.

Filipe Mesquita, Matteo Cannaviccio, Jordan Schmidek, Paramita Mirza, and Denilson Barbosa. 2019. KnowledgeNet: A benchmark dataset for knowledge base population. 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 749–758.

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.

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

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*, pages 3661–3672.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *ECML PKDD*, pages 148–163.

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

Changzhi Sun, Yeyun Gong, Yuanbin Wu, Ming Gong, Daxin Jiang, Man Lan, Shiliang Sun, and Nan Duan. 2019. Joint type inference on entities and relations via graph convolutional networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1361–1370.

Changzhi Sun, Yuanbin Wu, Man Lan, Shiliang Sun, Wenting Wang, Kuang-Chih Lee, and Kewen Wu. 2018. Extracting entities and relations with joint minimum risk training. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2256–2265.

Kai Sun, Richong Zhang, Samuel Mensah, Yongyi Mao, and Xudong Liu. 2020. Recurrent interaction network for jointly extracting entities and classifying relation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 3722–3732.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Yucheng Wang, Bowen Yu, Yueyang Zhang, Tingwen Liu, Hongsong Zhu, and Limin Sun. 2020. 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.

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.

Lin Xu, Qixian Zhou, Ke Gong, Xiaodan Liang, Jianheng Tang, and Liang Lin. 2019. End-to-end knowledge-routed relational dialogue system for automatic diagnosis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7346–7353.

Hongbin Ye, Ningyu Zhang, Shumin Deng, Mosha Chen, Chuanqi Tan, Fei Huang, and Huajun Chen. 2021. Contrastive triple extraction with generative transformer. In *Proceedings of the 35th AAAI Conference on Arificial Intelligence*.

Bowen Yu, Zhenyu Zhang, Xiaobo Shu, Yubin Wang, Tingwen Liu, Bin Wang, and Sujian Li. 2020. Joint extraction of entities and relations based on a novel decomposition strategy. In *Proc. of ECAI*.

Mo Yu, Wenpeng Yin, Kazi Saidul Hasan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2017. Improved neural relation detection for knowledge base question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 571–581.

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 29th International Joint Conference on Arificial Intelligence*, pages 4054–4060.

Dmitry Zelenko, Chinatsu Aone, and Richardella Anthony. 2002. Kernel methods for relation extraction. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 71–78.

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 34th AAAI Conference on Arificial Intelligence*.

Xiangrong Zeng, Shizhu He, Daojian Zeng, Kang Liu, Shengping Liu, and Jun Zhao. 2019. Learning the extraction order of multiple relational facts in a sentence with reinforcement learning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 367–377.

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.

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.

GuoDong Zhou, Jian Su, Jie Zhang, and Min Zhang. 2005. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 427–434, Philadelphia. Association for Computational Linguistics.