Synergetic Interaction Network with Cross-task Attention for Joint Relational Triple Extraction

Da Luo*, **Run Lin***, **Qiao Liu**†, **Yuxiang Cai, Xueyi Liu, Yanglei Gan, Rui Hou**
University of Electronic Science and Technology of China
{luoda, runlin, yuxiangcai, xueyiliu, yangleigan, hour}@std.uestc.edu.cn, qliu@uestc.edu.cn

Abstract

Joint entity-relation extraction remains a challenging task in information retrieval, given the intrinsic difficulty in modelling the interdependence between named entity recognition (NER) and relation extraction (RE) sub-tasks. Most existing joint extraction models encode entity and relation features in a sequential or parallel manner, allowing for limited one-way interaction. However, it is not yet clear how to capture the interdependence between these two sub-tasks in a synergistic and mutually reinforcing fashion. With this in mind, we propose a novel approach for joint entity-relation extraction, named **S**ynergetic **I**nteraction **Net**work (SINET) which utilizes a cross-task attention mechanism to effectively leverage contextual associations between NER and RE. Specifically, we construct two sets of distinct token representations for NER and RE sub-tasks respectively. Then, both sets of unique representation interact with one another via a cross-task attention mechanism, which exploits associated contextual information produced by concerted efforts of both NER and RE. Experiments on three benchmark datasets demonstrate that the proposed model achieves significantly better performance in joint entity-relation extraction. Moreover, extended analysis validates that the proposed mechanism can indeed leverage the semantic information produced by NER and RE sub-tasks to boost one another in a complementary way. The source code is available to the public online (https://github.com/AONE-NLP/RTE-SINET).

Keywords: relational triple extraction, named entity recognition, cross-task attention, interactive modeling

1. Introduction

The task of joint entity and relation extraction involves the simultaneous identification of entity mentions and their associated relations from unstructured textual data, with the goal of generating relational triples in the form of <subject, relation, object>. These extracted triples have been fundamental in a wide range of downstream applications such as knowledge base construction (Luan et al., 2018; Distiawan et al., 2019), question answering (Li et al., 2019; Xu et al., 2016) and relational reasoning (Santoro et al., 2017; Zhu et al., 2019).

In the realm of relation extraction (RE), conventional approach relied on a pipeline-based framework, where all entities in a sentence are first extracted and subsequently used to classify the relation between all entity pairs (Zhou et al., 2005; Chan and Roth, 2011). This approach was initially favored due to its simplicity and ease of implementation. However, it fails to consider the interdependence between named entity recognition (NER) and RE, leading to error propagation.

Recent studies have explored alternative strategies, such as sequential or parallel encoding, to jointly capture the interaction between NER and RE (Eberts and Ulges, 2020; Lu et al., 2022). Sequential encoding (Bekoulis et al., 2018b; Wei et al., 2020; Bekoulis et al., 2018a) involves performing NER and RE sequentially or vice versa, with the

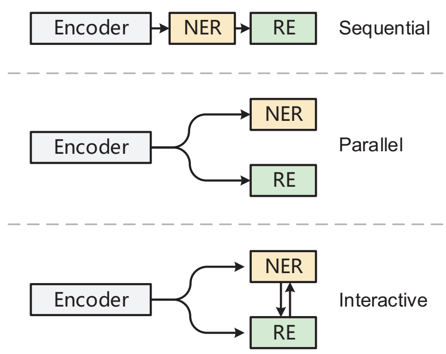


Figure 1: Three interactive learning structures for joint entity-relation extraction, where the directed arrows signify the flow of information between NER and RE sub-tasks.

output of one sub-task serving as the input to the other as illustrated in Figures 1; however, these approaches suffer from error accumulation due to its uni-directional interaction between both sub-tasks. Conversely, parallel encoding (Fu et al., 2019; Wadden et al., 2019) generates task-specific features independently via the same input. While parallel encoding strategies could alleviate the issue of error accumulation, they risk insufficient interaction between the NER and RE sub-tasks which may hinder to model important nuances and relationships between entities and relations, as their interaction is limited to input sharing.

To fill this gap, recent studies have investigated two-way interactive encoding methods that can capture

* Equal contribution
† Corresponding author

LREC-COLING 2024, pages 15457–15468
20-25 May, 2024.

the implicit interaction between NER and RE sub-tasks. A recurrent interaction architecture has been proposed to dynamically capture the correlation between NER and RE (Sun et al., 2020; Wang and Lu, 2020; Yan et al., 2021; Xiong et al., 2022). These approaches utilize a recurrent network (e.g. LSTM) to learn the interdependencies between the NER and RE, enabling dynamical inner-task interaction. However, such a recurrent learning paradigm processes the entire sequence of both sub-tasks at each time step, entailing the integration of multiple gate units and recursive computations, consequently leading to a significant computational burden.(Williams et al., 2015).

In this paper, we propose a novel joint relation extraction framework that captures synergetic interaction between NER and RE by exploiting associated contextual information produced from both sub-tasks, as inspired by the work (Chen et al., 2021). More precisely, we introduce a **c**ross-**t**ask **a**ttention **m**echanism (CTAM) that coordinates the back-propagation between NER and RE to synchronize their attention allocation and promote effective synergy. Given a sentence, SINET first transforms its pre-trained embeddings into task-aware spaces (Ren et al., 2022) corresponding to NER and RE, respectively, with each having its own task-aware projection function. Subsequently, SINET employs the CTAM to exchange information among two sub-tasks in a mutual manner, thereby obtaining contextual associated representations for both, respectively. These contextual representations serve as the agent vectors in the subsequent NER and RE sub-tasks, and are used to tune attention allocation of each sub-task during back-propagation.

Experimental results show that, comparing to other joint-learning models, the proposed SINET model can better capture semantic contextual association between entity pairs and relational facts on benchmark trials. In addition, SINET is more efficient than current baselines, as it is fast on decoding and requires less computation resources. Our contributions can be summarized as:

- We present a novel synergetic interaction network that enables effective bi-directional interaction between NER and RE sub-tasks by leveraging contextual association.
- Our proposed method employs a cross-task attention mechanism that facilitates synergetic interaction between NER and RE sub-tasks. This mechanism exploits the potential contextual association between NER and RE to enhance contextual understanding and inference abilities of the joint learning model with a linear computation and memory.
- We conduct extensive experiments on three standard benchmarks. The experimental results indicate that our method performs better than state-of-the-art baselines.

2. Related Work

Relational triple extraction is a task which typically involves two sub-tasks: named entity recognition (NER) and relation extraction (RE). Traditional pipelined methods (Chan and Roth, 2011; Gormley et al., 2015) performed relation extraction by considering all possible combinations of extracted entity pairs. However, such approaches neglect the interdependence of NER and RE sub-tasks. To exploit interrelated signals between two sub-tasks, feature-based joint models (Miwa and Sasaki, 2014; Ren et al., 2017) have been proposed. However, these models are based on manual operations and Natural Language Processing (NLP) tools, which are prone to data labeling errors.

Since the introduction of recursive neural network for RE by (Socher et al., 2012), a variety of sequence tagging models have been investigated (Miwa and Bansal, 2016; Zheng et al., 2017). Although promising results have been achieved, these methods are unable to handle overlapping entities. To overcome this limitation, (Dai et al., 2019) introduced a position-attention mechanism to generate various sentence representations with respect to every query position, facilitating the extraction of overlapping entities. Similarly, (Yu et al.) proposed a unified sequence labeling framework based on a novel decomposition strategy. However, these approaches remain insufficient in dealing with EntityPairOverlap (EPO) triples.

Recent research has focused on inter-task interactive modeling to address this issue. For instance, (Bekoulis et al., 2018b) proposed a multi-head selection method to predict potential relations of each entity pair. (Wei et al., 2020) proposed a novel cascade binary tagging scheme to obtain entities and relations, and span-based models (Eberts and Ulges, 2020; Ji et al., 2020) have been utilized for joint relation extraction, substantially reducing training complexity through negative sampling strategies. However, these methods encode entity and relation information sequentially, resulting in a disparity of information exposure between NER and RE sub-tasks.

Another trend of research has been devoted to the parallel encoding of task-specific features, which involves input sharing to facilitate the interaction between sub-tasks. For example, (Fu et al., 2019) encoded entity and relation information separately by using common features derived from graph convolutional networkencoder. More recently, A dynamic graph information extraction method (DYGIE) (Luan et al., 2019) have utilized dynamic graphs of spans to enhance interaction across NER and RE sub-tasks, enabling the model to learn useful information from a broader context. Moreover, (Wadden et al., 2019) reinforced DyGIE by implementing a transformer architecture from (Devlin et al., 2019) to capture task-independent textual context.

Despite the initial success of the aforementioned approaches, they have not adequately addressed the challenge of modeling two-way interaction between NER and RE sub-tasks. (Sun et al., 2020) proposed a multi-task learning method to dynamically learn the interactions between two sub-tasks, while (Wang and Lu, 2020) proposed the table-sequence encoders where a sequence encoder and a table encoder are designed to help each other and complete explicit interactions. Besides, a partition filter network (Yan et al., 2021) was proposed to handle inter-task communication via the shared contextual partition. However, those models process the entire sequence at each step and impose a substantial computational load. In our proposed SINET, we leverage CTAM to effectively utilize the contextual association produced from synergetic interaction between NER and RE, thereby ensuring a linear time complexity for computation and memory utilization.

3. Method

3.1. Problem Definition

Given a sentence X consisting of *n* tokens $\{x_1, x_2, ..., x_n\}$. The desired outputs are denoted as $Y(X) = \{(s, r, o) \mid s, o \in E, r \in R\}$, where *E* and *R* are the pre-defined entity and relation sets, respectively. To help introduce the rest, all possible spans from the sentence are defined as $S = \{s_1, s_2, ..., s_m\}$, where m denotes the total number of all possible spans.

3.2. Our Approach

The overall structure of SINET is depicted in Figure 2, which has five main components, i.e., a BERT encoder module, a task-aware feature projection module (TAP for short), a cross-task attention mechanism module (CTAM for short), an entity classifier module, and a multi-label relation extraction module. Benefiting from the TAP and CTAM module, entity classifier module and relation extraction module interact in a multi-task learning manner during the training phase, but operate sequentially in reasoning phase.

BERT Encoder We map word embeddings into BERT embeddings via pre-trained transformer blocks (Vaswani et al., 2017). As in the original implementation BERT (Devlin et al., 2019), an additional classification token CLS is added to the sequence, which acts as an agent that summarizes all the tokens. Hence, the unified embedding sequence obtained from BERT can be denoted as $Y_{enc}(X) = \{\mathbf{h}_{cls}, \mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n \mid \mathbf{h}_i \in \mathbb{R}^{d\times 1}\}$, where $\mathbf{h}_i$ represents the hidden state of the i-th token in the input sequence, and d is the dimension of the hidden state.

Task-aware Feature Projection We view both NER and RE sub-tasks from the multi-task feature learning perspective. Intuitively, NER seeks to extract entity mention (mainly entity type) knowledge from a set of labeled training data; whereas RE aims to learn the distribution of entity mentions driven by the factual relation types. Thus, rather than adopting the unified semantic representation from BERT for both sub-tasks, the key insight is to learn a projection from the semantic space to the sub-task feature space. More specifically, the generic sentence representation is adopted from BERT, and then fed into a set of isomorphic *task-aware feature projection* (TAP) layers for NER and RE respectively. The TAP layers are composed of trainable matrices denoted by $\mathbf{W}^e$ and $\mathbf{W}^r$, with dimensions of $\mathbb{R}^{d\times d}$, respectively. The TAP with respect to both sub-tasks are defined as,

$$\mathbf{h}_i^e = f^e(\mathbf{h}_i) = \mathbf{W}^e\mathbf{h}_i, \mathbf{h}_i^r = f^r(\mathbf{h}_i) = \mathbf{W}^r\mathbf{h}_i. \quad (1)$$

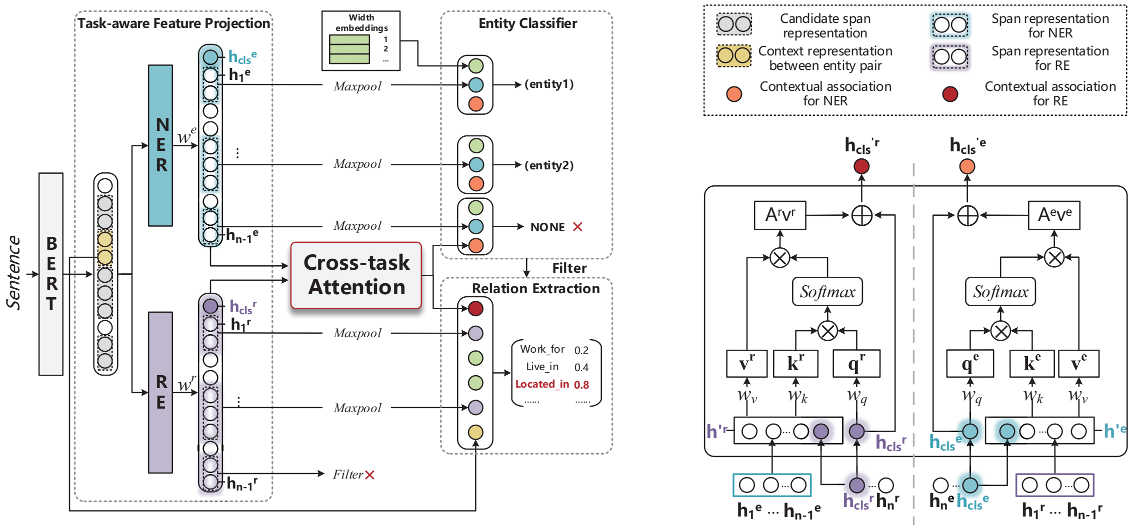
The output of TAP denotes as *task-aware representation* in the sense that the feature representation of the same input can be varying under different sub-task specifications.

Cross-task Attention Mechanism The cross-task learning between NER and RE involves the CLS token of one sub-task feature space and *task-aware representation* of the other feature space. Specifically, in order to interact the task-aware features more effectively and efficiently, the CLS token is deployed as an agent to exchange information among the sequence of tokens from the other feature space.

As illustrated in Figure 2 (b), entity-aware feature representations and relation-aware feature representations are concerted with one another to mutually model the contextual associations between NER and RE via *Multi-head Cross-task Attention Mechanism*. Specifically, the CTAM takes *task-aware representation* of NER and RE as input, for entity-aware space, it gathers *task-aware representation* from relation space and concatenates its own CLS token to them and the same procedure is applied to the relation-aware space by simply swapping the index *e* and *r*, denoted as:

$$\mathbf{h}'^e = [\mathbf{h}_i^r; \mathbf{h}_{cls}^e], \mathbf{h}'^r = [\mathbf{h}_i^e; \mathbf{h}_{cls}^r]. \quad (2)$$

We then conduct CTAM between $\mathbf{h}_{cls}^e$ and $\mathbf{h}'^e$, $\mathbf{h}_{cls}^r$ and $\mathbf{h}'^r$, where CLS token acts as the query vector. Following the original transformer implementation, multiple attention heads are also deployed in CTAM,



(a) Framework of SINET (b) Cross-task Attention Mechanism

Figure 2: (a) Overview of SINET. The framework consists of five components: BERT encoder, task-aware feature projection module, cross-task attention mechanism module, entity classifier and relation extraction module. Cyan and orchid denote the two task-aware spaces respectively. Orange and red denote the contextual association captured by CTAM for NER and RE respectively, the yellow block denote the context representation between a pair entities (local context for short), and green blocks denote the span width embedding. (b) Detailed depiction of CTAM. The CLS token of the entity-aware space serves as a query token to interact with tokens from the relation-aware space. The same procedure applies to the relation-aware space when interacting with entity-aware space.

denoted as:

$$\begin{aligned}
\mathbf{q}^e &= \mathbf{W}_q\mathbf{h}^e_{cls}, \mathbf{k}^e = \mathbf{W}_k\mathbf{h}'^e, \mathbf{v}^e = \mathbf{W}_v\mathbf{h}'^e, \\
\mathbf{q}^r &= \mathbf{W}_q\mathbf{h}^r_{cls}, \mathbf{k}^r = \mathbf{W}_k\mathbf{h}'^r, \mathbf{v}^r = \mathbf{W}_v\mathbf{h}'^r, \\
\mathbf{A}^e &= softmax(\mathbf{q}^e\mathbf{k}^{eT}/\sqrt{d/h}), \\
\mathbf{A}^r &= softmax(\mathbf{q}^r\mathbf{k}^{rT}/\sqrt{d/h}), \\
\mathbf{h}'^e_{cls} &= \mathbf{A}^e\mathbf{v}^e + \mathbf{h}^e_{cls}, \mathbf{h}'^r_{cls} = \mathbf{A}^r\mathbf{v}^r + \mathbf{h}^r_{cls},
\end{aligned} \tag{3}$$

where $\mathbf{W}_q$, $\mathbf{W}_k$, $\mathbf{W}_v \in \mathbb{R}^{d\times(d/h)}$ are learnable parameters, d and h denote the embedding dimension and number of attention heads respectively. The outputs of the cross-task attention module with layer normalization (LN) are defined as:

$$\mathbf{h}'^e_{cls} = LN(\mathbf{h}'^e_{cls}), \mathbf{h}'^r_{cls} = LN(\mathbf{h}'^r_{cls}). \tag{4}$$

Entity Classifier As Figure 2 (a) shows, entity classifier takes three components as input, which comprises the representation of the entity span via TAP, the width embedding of the span, and the context-associated CLS for NER obtained from CTAM.

- The representation of entity span is denoted as $\mathbf{S}^e = \{\mathbf{s}^e_1, \mathbf{s}^e_2, ..., \mathbf{s}^e_m\}$. All tokens in each span are integrated using max pooling to get the entity span representation $\mathbf{s}^e_i$:

$$\mathbf{s}^e_i = Maxpool\left\{\mathbf{h}^e_i, \mathbf{h}^e_{i+1}, ..., \mathbf{h}^e_j\right\}. \tag{5}$$

- The width embedding $\mathbf{w}_k$ is generated by a dedicated embedding matrix, which is learned through back-propagation, based on the span width k.

- The associated context representation for NER, $\mathbf{h}'^e_{cls}$, is obtained from the interaction between NER and RE via CTAM.

The final representation for the entity classifier is obtained by concatenating these three inputs:

$$\mathbf{e}_i = [\mathbf{s}^e_i; \mathbf{w}_k; \mathbf{h}'^e_{cls}], \tag{6}$$

The final representation $\mathbf{e}_i$ is then fed to a Feed-forward Neural Network (FNN) followed by a Softmax classifier in order to determine corresponding entity class, denoted as:

$$\mathbf{y}^e = Softmax(\mathbf{W}_{ner}\mathbf{e}_i + \mathbf{b}^e), \tag{7}$$

where $\mathbf{y}^e$ is the posterior for each entity class (including *none* type). And we exclude spans that are predicted as *none* type, and form a candidate entity set $\mathbf{S}^r = \{\mathbf{s}^r_1, \mathbf{s}^r_2, ..., \mathbf{s}^r_{m'}\}$ $(m' \leq m)$ in order to reduce searching space when performing RE.

Multi-label Relation Extraction The relation classifier examines each candidate pair of entities and determines if any relations from the pre-defined relation set *R* present. Figure 2 (a) illustrates the three components that the relation classifier takes as input, including the candidate entity tuple representations along with their width embeddings, local context representation and the associated context representation for RE obtained from CTAM.

- The relation span's embeddings are denoted as $\mathbf{S}^r = \{\mathbf{s}_1^r, \mathbf{s}_2^r, ..., \mathbf{s}_{m'}^r\}$. Given two entity mentions s_1, s_2 and their corresponding widths k, we concatenate the relation span representation $\mathbf{s}_i^r$ and width embedding $\mathbf{w}_k$ to represent the entity candidates $\mathbf{e}(s_1)$, $\mathbf{e}(s_2)$. The detailed operations are as follows:

$$\begin{aligned} \mathbf{s}_i^r &= Maxpool\left\{\mathbf{h}_i^r, \mathbf{h}_{i+1}^r, ..., \mathbf{h}_j^r\right\}, \\ \mathbf{e}(s_1) &= [\mathbf{s}_1^r; \mathbf{w}_k], \mathbf{e}(s_2) = [\mathbf{s}_2^r; \mathbf{w}_k]. \end{aligned} \tag{8}$$

- The local context consists of the tokens between two potential entity spans. We leverage their BERT embeddings and obtain the local context representation $\mathbf{c}(s_1, s_2)$ through max pooling. If any pairs of entity spans are contiguous, we set $\mathbf{c}(s_1, s_2) = \mathbf{0}$.

- The associated context representation for RE is denoted as $\mathbf{h}_{cls}^{\prime r}$ and is obtained from CTAM.

We concatenate these four embeddings to form the final representation for the relation classifier. Since relations are asymmetric in general, we need to classify both (s_1, s_2) and (s_2, s_1):

$$\begin{aligned} \mathbf{r_1} &= [\mathbf{e}(s_1); \mathbf{c}(s_1, s_2); \mathbf{e}(s_2); \mathbf{h}_{cls}^{\prime r}] \\ \mathbf{r_2} &= [\mathbf{e}(s_1); \mathbf{c}(s_2, s_1); \mathbf{e}(s_2); \mathbf{h}_{cls}^{\prime r}] \end{aligned} \tag{9}$$

which is then passed to a multi-label relation classifier, denoted as:

$$\mathbf{y}^r = Sigmoid(\mathbf{W}_{re}\mathbf{r_{1/2}} + \mathbf{b}^r), \tag{10}$$

where $\mathbf{y}^r$ is the predicted score of each relation type. If the score exceeds the relation filtering threshold α (see section 4.2 for details), it implies that the corresponding relations exist between s_1 and s_2.

3.3. Training Strategy

The joint loss function for NER and RE is denoted as:

$$\mathcal{L} = \mathcal{L}^e + \mathcal{L}^r \tag{11}$$

where $\mathcal{L}^e$ is the cross-entropy loss for NER and $\mathcal{L}^r$ is the binary cross-entropy loss for RE.
Note that we use the same negative sampling strategy as (Eberts and Ulges, 2020). For entity classifier, we select entity spans that have been tagged as positive samples, and randomly choose negative samples in the rest of spans. For relation classifier, we use ground truth relations as positive samples, and draw negative samples from those entity pairs that are not labeled with any relations.

Dataset	#Sentence Train	Dev	Test	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{R}\rvert$
ACE05	10,051	2,424	2,050	7	6
ACE04	8,683(5-fold)			7	6
SciERC	1,861	275	551	6	7

Table 1: The statistics for datasets ACE05, ACE04 and SciERC . $|\mathcal{E}|$ and $|\mathcal{R}|$ are numbers of entity and relation types, respectively.

4. Experiments

4.1. Datasets and Evaluation Metrics

We evaluate our approach on three datasets: ACE04, ACE05 (Walker et al., 2005), SciERC (Luan et al., 2018). The ACE04 and ACE05 datasets are collected from various sources, such as news articles and online forums. The SciERC dataset is collected from 500 AI paper abstracts. We follow the preprocessing steps in DYGIE[1] for ACE04[2] and ACE05[3]. Specifically, for ACE04, we use 5-fold cross validation to evaluate the model, and the development set construct by 15% of training set. Table 1 shows the data statistics.
Furthermore, we adopt the evaluation protocol used in prior works and measure the performance of our model using micro Precision, Recall and F1-score as the evaluation metric. For NER, a correct entity prediction must have both correct type and boundary. For RE, we adopt two evaluation metrics: (1) *boundaries* evaluation (Rel): considers a triple prediction correct only if the predicted relation and the boundaries of two spans are correct. (2) *strict* evaluation (Rel+): requires predicted entity types to be correct in addition to satisfying the conditions of the *boundaries* evaluation.

4.2. Implementation Details

For fair comparison, we use *bert-base-cased* (Devlin et al., 2019) and *albert-xxlarge-v1* (Lan et al., 2019) as the base encoders for ACE04 and ACE05, *scibert-scivocab-cased* (Beltagy et al., 2019) for SciERC. The batch size and learning rate are set to 4/20 and $1e^{-5}/2e^{-5}$ for SciERC/others respectively. We also conduct a dropout before the entity and relation classifier with a rate of 0.1 and width embeddings size w of 100. For CTAM, the attention head is set to 4 with a dropout rate of 0.2. In addition, the max span size and relation filtering threshold α are set to 8 and 0.5. The layers of CTAM is set to 1. The model is trained on a single NVIDIA A100 Tensor Core GPU and an Intel Xeon Bronze 3104 CPU for 100 epochs on all three datasets. Furthermore, we report the averaged F1-scores of 5 runs to ensure the robustness of our results.

[1] https://github.com/luanyi/DyGIE
[2] https://catalog.ldc.upenn.edu/LDC2005T09
[3] https://catalog.ldc.upenn.edu/LDC2006T06

Model	Encoder	ACE04			ACE05			SciERC		
		Ent	Rel	Rel+	Ent	Rel	Rel+	Ent	Rel	Rel+
SPTree (Miwa and Bansal, 2016)	Bb	81.80	-	48.40	83.40	-	55.60	-	-	-
SciIE (Luan et al., 2018)	SciB	-	-	-	-	-	-	64.20	39.30	-
DYGIE (Luan et al., 2019)	Bb	87.40	59.70	-	-	-	-	-	-	-
DYGIE++ (Wadden et al., 2019)	Bb	-	-	-	88.60	63.40	-	-	-	-
DYGIE++ (Wadden et al., 2019)	SciB	-	-	-	-	-	-	67.50	48.40	-
Multi-turn QA (Li et al., 2019)	Bb	83.60	-	49.40	84.80	-	60.20	-	-	-
Two are Better than One (Wang and Lu, 2020)	ALB	88.60	-	59.60	89.50	-	64.30	-	-	-
SPE (Wang et al., 2020)	SciB	-	-	-	-	-	-	66.90	-	33.60
SpERT (Eberts and Ulges, 2020)	SciB	-	-	-	-	-	-	70.33	50.84	-
SPAN$_{Multi-Head}$ (Ji et al., 2020)	Bb	-	-	-	89.59	65.24	-	-	-	-
PURE (Zhong and Chen, 2021)	ALB	88.80	64.70	60.20	89.70	69.00	65.60	-	-	-
PURE (Zhong and Chen, 2021)	SciB	-	-	-	-	-	-	66.60	48.20	35.60
PFN (Yan et al., 2021)	ALB	89.30	-	62.50	89.00	-	66.80	-	-	-
PFN (Yan et al., 2021)	SciB	-	-	-	-	-	-	66.80	-	38.40
MGE (Xiong et al., 2022)	ALB	89.30	-	63.80	89.70	-	**68.20**	-	-	-
MGE (Xiong et al., 2022)	SciB	-	-	-	-	-	-	68.40	-	39.40
	Bb	<u>88.27</u>	<u>60.86</u>	<u>57.34</u>	<u>88.58</u>	<u>66.18</u>	<u>62.83</u>	-	-	-
SINET (Ours)	SciB	-	-	-	-	-	-	**<u>72.59</u>**	**<u>51.01</u>**	**<u>40.13</u>**
	ALB	**<u>90.53</u>**	**<u>66.53</u>**	**<u>64.65</u>**	**<u>90.56</u>**	**<u>69.04</u>**	<u>65.71</u>	-	-	-

Table 2: Comparison (%) of the proposed SINET with the prior works on three datasets. **Bold** represents the best F1-scores. Results that indicate a statistically significant improvement with a p-value of less than **0.05** under the bootstrap paired t-test are marked with an <u>underline</u>. Bb, ALB, SciB denote the use of Bert-base, ALBERT-xxlarge-v1 and SciBERT encoders, respectively.

5. Main Results

As shown in Table 2, we conduct a comprehensive evaluation of SINET against prior state-of-the-art models on three publicly available datasets (ACE04, ACE05, SciERC). The experimental results demonstrate the superiority of our proposed method over prior SOTA works in the field of NER. Moreover, our model also achieves promising performance in RE. These findings highlight the effectiveness of our proposed method SINET, which reinforces the significance of modeling synergetic interaction between entity pairs and relational facts in joint entity-relation extraction task.

Regarding NER, our model exhibits an improvement on F1-scores of +1.23%, +0.85%, +2.26% on ACE04, ACE05 and SciERC, respectively, compared to the previous methods. As for RE, our model outperforms previous baselines on all three datasets. Under the $boundaries$ evaluation standard, our model achieves an absolute improvement on F1-scores of +1.83%, +0.04%, +0.17% on ACE04, ACE05, and SciERC respectively, when compared to the previous baseline methods. Furthermore, under the $strict$ evaluation standard, our model surpasses the previous baselines on F1-scores by a margin of +0.85%, +0.73% on ACE04 and SciERC respectively. We also noticed that compared to the previous SOTA models based on Bert-base, our model achieves a substantially better relation F1. Overall, our proposed method outperforms previous baselines, achieving a remarkable improvement in performance.

6. Analysis

6.1. Ablation Study

In this section, we provide an in-depth examination of the efficacy of our framework in NER and RE, appraising it based on five distinct aspects:

Attention head numbers. We conduct comparative experiments to investigate the impact of the number of attention heads in CTAM on the SciERC development set. As shown in Figure 3, optimal performance of the model is achieved with 4 attention heads. However, increasing the number of attention heads beyond 4 did not yield any significant improvements in either NER or RE sub-tasks.

CTAM layers. In accordance with the results presented in Figure 3, our model adopts a multi-layer architecture, with each layer composed of a CTAM module. Upon evaluating the CTAM depths (D) of one to five layers, we observe a drop in performance of approximately 0.82%, 1.28% and 1.63% F1-scores for entity, $boundaries$ relation and $strict$ relation extraction, respectively. We argue that excessive layers increases the number of model parameters, which may lead to over-fitting and ultimately hinder the model's performance.

Entity span representation. In terms of entity span representation, we explored various alternatives, including max pooling, sum pooling, and average pooling over entity tokens. Our findings, detailed in Table 3, indicate that averaging entity tokens is ineffective for both NER and RE. Instead, we found that sum pooling improves performance, achieving F1 scores of 73.84% and 40.66% for NER and "$strict$" RE, respectively. However, max

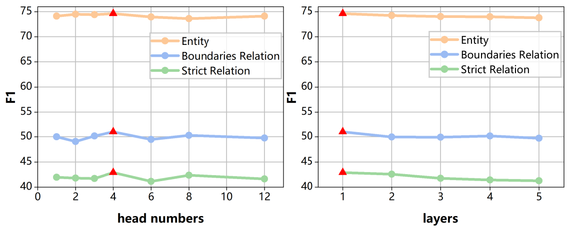


Figure 3: Performance of Proposed SINET with Varying Head Numbers and Layers of CTAM on the SciERC Development Set.

Ablation	Settings	Ent	Rel	Rel+
Entity Span $(\mathbf{s}_i^e, \mathbf{s}_i^r)$	Max	**74.65**	**50.99**	**42.91**
	Sum	73.84	48.80	40.66
	Average	71.33	44.14	35.71
Local Context $\mathbf{c}(s_i, s_j)$	Max	**74.65**	**50.99**	**42.91**
	Sum	73.75	48.38	41.02
	Average	73.66	45.69	37.75
Interactive agent $(\mathbf{h}_{cls}^e, \mathbf{h}_{cls}^r)$	CLS	**74.65**	**50.99**	**42.91**
	Max	74.54	49.74	41.54
	Average	74.18	49.83	41.79
LayerNorm	with	**74.65**	**50.99**	**42.91**
	without	73.84	50.13	41.41

Table 3: Ablation experiments on SciERC development set. Ent , Rel and Rel+ represent entity F1-scores, boundaries and strict relation F1-scores. The best results are marked in **bold**.

pooling yields even better results, outperforming sum pooling by an additional 0.81% and 2.25% for NER and "strict" RE, respectively.

Local context representation. Table 3 also reveals the impact of local context representations on RE within our proposed model. Our experimental findings suggest that utilizing max pooling to represent local context yields the best results.

Interactive agent. We investigate alternative options for the interactive agent in CTAM besides utilizing the CLS token, such as employing the max and average pooling of the sentence. As presented in Table 3, the model utilizing the CLS token as the agent attains the highest F1-scores on SciERC, indicating the efficacy of the CLS token in synthesizing the information of the current sub-task and passing it to the next one through CTAM.

Impact of layer normalization. Finally, we conduct an experiment to compare the F1-score with and without LN on the contextual association representations $(\mathbf{h}'^e_{cls}, \mathbf{h}'^r_{cls})$. Our results demonstrate that the representations with LN outperform those without, achieving an improvement of 1.50% in the $strict$ relation F1-score.

6.2. Effects of Synergetic Interaction

In order to validate the efficacy of the synergetic interaction for both NER and RE in our SINET, we conduct experiments by testing different encoder variations on SciERC and ACE05 development set:

	Settings	Ent	Rel	Rel+
SciERC	Base	73.99	50.49	41.85
	Base+TAP	74.48	50.54	41.94
	Base+TAP+CTAM$_{NER}$	74.26	50.36	42.70
	Base+TAP+CTAM$_{RE}$	74.01	49.84	41.63
	Base+TAP+CTAM	**74.65**	**50.99**	**42.91**
ACE05	Base	86.57	63.70	60.42
	Base+TAP	86.75	63.83	60.56
	Base+TAP+CTAM$_{NER}$	86.76	63.80	60.35
	Base+TAP+CTAM$_{RE}$	86.83	64.09	60.98
	Base+TAP+CTAM	**86.85**	**64.67**	**61.42**

Table 4: Ablation study on SciERC (SciBERT) and ACE05 (BERT-base) development set. Ent, Rel, and Rel+ represent entity F1-scores, boundaries and strict relation F1-scores. The best results are marked in **bold**.

Base : $\mathbf{h}_{cls}$, generated by the original BERT, is utilized as the shared representation for both NER and RE sub-tasks. The representation of entity candidates ($\mathbf{e}(s_i)$) and the shared representation $\mathbf{h}_{cls}$ are concatenated for NER sub-task. The representation of each pair of entity candidates ($\mathbf{e}(s_1)$, $\mathbf{e}(s_2)$), the local contextual representation $\mathbf{c}(s_1, s_2)$ between two potential entity spans and the $\mathbf{h}_{cls}$ representation are concatenated for RE sub-task.

Base+TAP : $\mathbf{h}^e_{cls}$ and $\mathbf{h}^r_{cls}$, generated by the task-aware projection layers, are utilized for NER and RE respectively.

Base + TAP + CTAM$_{NER}$: $\mathbf{h}'^e_{cls}$ obtained from CTAM, and $\mathbf{h}^r_{cls}$ obtained from TAP are leveraged for NER and RE respectively.

Base + TAP + CTAM$_{RE}$: $\mathbf{h}^e_{cls}$ obtained from TAP, and $\mathbf{h}'^r_{cls}$ obtained from CTAM are leveraged for NER and RE respectively.

Base + TAP + CTAM : $\mathbf{h}'^e_{cls}$ and $\mathbf{h}'^r_{cls}$ obtained from CTAM module are leveraged for NER and RE respectively.

The Base setting models the interaction between NER and RE using original CLS token from BERT encoder. Table 4 demonstrates that employing TAP based on the original CLS improves the model's performance compared to the base. We argue that sub-task interaction via direct input feature sharing may lead to the feature conflict issue, which degrades the overall performance. On the other hand, implementation of TAP provides a marginal improvement as it enables task-aware feature learning. Furthermore, we train our model in an unidirectional interaction setting, where CTAM$_{NER}$ and CTAM$_{RE}$ are integrated with the base model and TAP to perform joint entity-relation extraction, respectively, while utilizing information from the other sub-task. Experimental results indicate that these two variants achieve comparable performance with

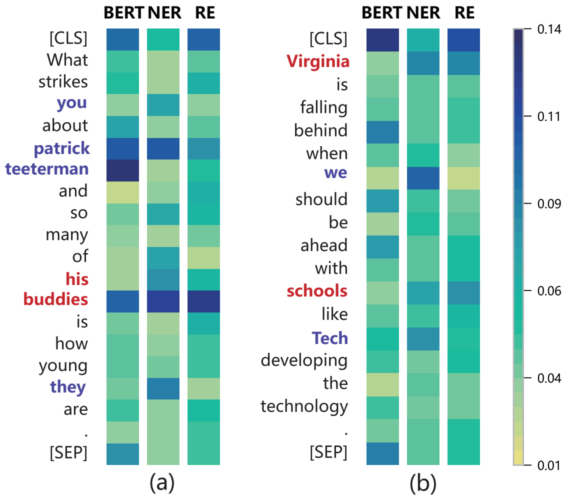


Figure 4: Attention heat maps of each token in the original BERT, NER-aware and RE-aware space via CTAM. The color intensity reflects the strength of the attention weight, with darker colors indicating higher weights. (Tokens marked in **bold** are gold entities, in which **red** denotes in-triple entities, and **blue** denotes out-of-triple entities).

"Base+TAP" model, which verifies the deficiency of unilateral interaction. Through a series of auxiliary experiments, we find that bi-directional interaction between NER and RE via TAP and CTAM leads to a mutual reinforcement of both sub-tasks and achieves the best performance.

6.3. Case Study

To further investigate the impact of interaction between NER and RE, we conduct extensive experiments and visualize the attention scores of each token on different semantic spaces with a couple of sample sentences from ACE05 dataset.

In Figure 4, it is observed that the original sentence representations from BERT indicate that most of BERT attention heads attend to the `CLS` token, which is intelligible since the class token is directly used for downstream tasks. However, the BERT sentence representations tend to overly focus on some of the gold entities (e.g., "`patrick teeterman`", "`buddies`") or place attention on task-irrelevant tokens (e.g., "`behind`", "`should`"). In our proposed SINET, the NER-aware space pays more attention to the gold entities, particularly the head token of an entity such as "patrick" in instance (a). And the RE-aware space places greater emphasis on in-triples entity pairs. As depicted in instance (b), entities marked in red ("`Virginia`" and "`schools`") that form an in-triple entity pair receive relatively higher attention in RE-aware space compared to other entities. Conversely, some out-of-triple entities (marked in blue), such as "`we`" and "`Tech`" receive relatively less attention in RE-aware space. Moreover, the attention scores are much more evenly distributed in RE-aware space than in the other two semantic spaces, highlighting the importance of global textual context in relation prediction. Overall, SINET uses the contextual association between NER and RE to its maximum advantage as it adaptively learns to interact with useful characteristics in a bi-directional and synergetic manner.

	Model	FLOPs (M)	Inference Time (s)
SciERC	SPAN_{Multi_Head}	3892.93	21
	PFN	1517.51	34
	SINET	**1279.25**	**11**
ACE05	Two are Better than One	**3867.13**	117
	PFN	26970.74	134
	SINET	26113.61	**65**

Table 5: Comparison of model efficiency on SciERC (SciBERT) and ACE05 (ALBERT-xxlarge-v1).

6.4. Model Efficiency

As shown in Table 5, computation efficiency of the proposed model is evaluated via computational complexity (**FLOPs**[4] and **inference time**) on both SciERC and ACE05 datasets. For the purpose of fair comparison, we follow the official implementations and default configurations of $\text{Span}_{Multi-Head}$(Ji et al., 2020), Two are Better Than One (Wang and Lu, 2020) and PFN (Yan et al., 2021). Batch size is set to 1 for all models when analyzing inference time.

The results indicate that the *FLOPs* of SINET is at least 15% lower than $\text{SPAN}_{Multi-Head}$ and PFN. During the inference phase, SINET is much quicker than $\text{SPAN}_{Multi-Head}$ and almost 1.5$\times$ speedup on PFN on decoding. Even though Two are Better Than One has lower **FLOPs** in the ACE05 dataset, SINET attains significant advantages on **inference time**, which confirms the efficiency of SINET.

7. Conclusion

In this paper, we present a novel synergetic interaction network for joint relational triple extraction model, named SINET. Instead of relying on shared information for both NER and RE, we employ a cross-task attention mechanism (CTAM), which takes full advantage of contextual association produced by NER and RE respectively, to explore the interaction between sub-tasks in a mutually reinforcing way. We conduct extensive experiments on three benchmark datasets to validate the effectiveness of our model. Overall results demonstrate that our method is superior to previous baselines in both NER and RE tasks. Auxiliary experiments show the significance of our model in dealing with synergetic interaction between NER and RE.

[4] https://github.com/Lyken17/pytorch-OpCounter

8. Acknowledgements

We would like to thank the anonymous reviewers for their valuable discussion and constructive feedback. This work is supported by the National Natural Science Foundation of China (U22B2061), the National Key R&D Program of China (2022YFB4300603) and Sichuan Science and Technology Program (2023YFG0151).

9. Bibliographical References

Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. 2018a. Adversarial training for multi-context joint entity and relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2830–2836.

Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. 2018b. Joint entity recognition and relation extraction as a multi-head selection problem. *Expert Systems with Applications*, 114:34–45.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 3615–3620.

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.

Chun-Fu Richard Chen, Quanfu Fan, and Rameswar Panda. 2021. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 357–366.

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

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

Bayu Distiawan, Gerhard Weikum, Jianzhong Qi, and Rui Zhang. 2019. Neural relation extraction for knowledge base enrichment. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 229–240.

Liang Du and Haibin Ling. 2015. Cross-age face verification by coordinating with cross-face age verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2329–2338.

Markus Eberts and Adrian Ulges. 2020. Span-based joint entity and relation extraction with transformer pre-training. In *Proceedings of the 24th European Conference on Artificial Intelligence*, pages 2006–2013.

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.

Matthew R Gormley, Mo Yu, and Mark Dredze. 2015. Improved relation extraction with feature-rich compositional embedding models. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1774–1784.

Luheng He, Kenton Lee, Omer Levy, and Luke Zettlemoyer. 2018. Jointly predicting predicates and arguments in neural semantic role labeling. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 364–369.

Bin Ji, Jie Yu, Shasha Li, Jun Ma, Qingbo Wu, Yusong Tan, and Huijun Liu. 2020. Span-based joint entity and relation extraction with attention-based span-specific and contextual semantic representations. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 88–99.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

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

Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2022.

Unified structure generation for universal information extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5755–5772.

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

Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. 2019. A general framework for information extraction using dynamic span graphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3036–3046.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011.

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.

Makoto Miwa and Yutaka Sasaki. 2014. Modeling joint entity and relation extraction with table representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing*, pages 1858–1869.

P Molchanov, S Tyree, T Karras, T Aila, and J Kautz. 2019. Pruning convolutional neural networks for resource efficient inference. In *5th International Conference on Learning Representations, ICLR 2017-Conference Track Proceedings*.

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

Feiliang Ren, Longhui Zhang, Xiaofeng Zhao, Shujuan Yin, Shilei Liu, and Bochao Li. 2022. A simple but effective bidirectional framework for relational triple extraction. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 824–832.

Xiang Ren, Zeqiu Wu, Wenqi He, Meng Qu, Clare R Voss, Heng Ji, Tarek F Abdelzaher, and Jiawei Han. 2017. Cotype: Joint extraction of typed entities and relations with knowledge bases. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1015–1024.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 74–84.

Dan Roth and Wen-tau Yih. 2004. A linear programming formulation for global inference in natural language tasks. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004*, pages 1–8.

Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. 2017. A simple neural network module for relational reasoning. *Advances in neural information processing systems*, 30.

Ozan Sener and Vladlen Koltun. 2018. Multi-task learning as multi-objective optimization. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 525–536.

Herbert A Simon. 1983. Why should machines learn? In *Machine learning*, pages 25–37. Elsevier.

Richard Socher, Brody Huval, Christopher D Manning, and Andrew Y Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 1201–1211.

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

Liwen Sun, Michael J Franklin, Jiannan Wang, and Eugene Wu. 2016. Skipping-oriented partitioning for columnar layouts. *Proceedings of the VLDB Endowment*, 10(4):421–432.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 5784–5789.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2005. Ace 2005 multilingual training corpus-linguistic data consortium. *URL: https://catalog. ldc. upenn. edu/LDC2006T06*.

Jue Wang and Wei Lu. 2020. Two are better than one: Joint entity and relation extraction with table-sequence encoders. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Yijun Wang, Changzhi Sun, Yuanbin Wu, Junchi Yan, Peng Gao, and Guotong Xie. 2020. Pre-training entity relation encoder with intra-span and inter-span information. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 1692–1705.

Yiwei Wang, Muhao Chen, Wenxuan Zhou, Yujun Cai, Yuxuan Liang, Dayiheng Liu, Baosong Yang, Juncheng Liu, and Bryan Hooi. 2022. Should we rely on entity mentions for relation extraction? debiasing relation extraction with counterfactual analysis. pages 3071–3081.

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.

Will Williams, Niranjani Prasad, David Mrva, Tom Ash, and Tony Robinson. 2015. Scaling recurrent neural network language models. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5391–5395. IEEE.

Hui Wu and Xiaodong Shi. 2021. Synchronous dual network with cross-type attention for joint entity and relation extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2769–2779.

Xiong Xiong, Yunfei Liu, Anqi Liu, Shuai Gong, and Shengyang Li. 2022. A multi-gate encoder for joint entity and relation extraction. In *China National Conference on Chinese Computational Linguistics*, pages 163–179. Springer.

Benfeng Xu, Quan Wang, Yajuan Lyu, Yabing Shi, Yong Zhu, Jie Gao, and Zhendong Mao. 2022. Emrel: Joint representation of entities and embedded relations for multi-triple extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 659–665.

Kun Xu, Siva Reddy, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2016. Question answering on freebase via relation extraction and textual evidence. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2326–2336.

Lu Xu, Hao Li, Wei Lu, and Lidong Bing. 2020. Position-aware tagging for aspect sentiment triplet extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2339–2349.

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

Deming Ye, Yankai Lin, Peng Li, and Maosong Sun. 2022. Packed levitated marker for entity and relation extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4904–4917.

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

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. 2020. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33:5824–5836.

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.

Yuhao Zhang, Peng Qi, and Christopher D Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2205–2215.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45.

Hengyi Zheng, Rui Wen, Xi Chen, Yifan Yang, Yunyan Zhang, Ziheng Zhang, Ningyu Zhang, Bin Qin, Xu Ming, and Yefeng Zheng. 2021. Prgc: Potential relation and global correspondence based joint relational triple extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6225–6235.

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.

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

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.

Hao Zhu, Yankai Lin, Zhiyuan Liu, Jie Fu, Tat-Seng Chua, and Maosong Sun. 2019. Graph neural networks with generated parameters for relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1331–1339.