# A Multi-Gate Encoder for Joint Entity and Relation Extraction

**Xiong Xiong**[1,2], **Yunfei Liu**[1,2], **Anqi Liu**[1], **Shuai Gong**[1,2], **Shengyang Li**[1,2*]
[1]Key Laboratory of Space Utilization, Technology and Engineering Center for Space Utilization,
Chinese Academy of Sciences, Beijing, China
[2]University of Chinese Academy of Sciences, Beijing, China
{xiongxiong20,liuyunfei,liuaq,gongshuai19,shyli}@csu.ac.cn

## Abstract

Named entity recognition and relation extraction are core sub-tasks of relational triple extraction. Recent studies have used parameter sharing or joint decoding to create interaction between these two tasks. However, ensuring the specificity of task-specific traits while the two tasks interact properly is a huge difficulty. We propose a multi-gate encoder that models bidirectional task interaction while keeping sufficient feature specificity based on gating mechanism in this paper. Precisely, we design two types of independent gates: task gates to generate task-specific features and interaction gates to generate instructive features to guide the opposite task. Our experiments show that our method increases the state-of-the-art (SOTA) relation F1 scores on ACE04, ACE05 and SciERC datasets to 63.8% (+1.3%), 68.2% (+1.4%), 39.4% (+1.0%), respectively, with higher inference speed over previous SOTA model.

## 1 Introduction

Extracting relational facts from unstructured texts is a fundamental task in information extraction. This task can be decomposed into two sub-tasks: Named Entity Recognition (NER) (Florian et al., 2003), which aims to recognize the boundaries and types of entities; and Relation Extraction (RE) (Zelenko et al., 2002), which aims to extract semantic relations between entities. The extracted relational triples in the form of (subject, relation, object) are basic elements of large-scale knowledge graphs (Lin et al., 2015).

Traditional approaches perform NER and RE in a pipelined fashion (Zhou et al., 2005; Chan and Roth, 2011; Gormley et al., 2015). They first extract all the entities in a given text, and then identify pairwise relations between the extracted entities. However, because the two sub-tasks are modeled independently, pipelined methods are vulnerable to error propagation issue. Since the interaction between NER and RE is neglected, the errors accumulated in the previous NER stage cannot be corrected in the subsequent RE stage. To resolve this issue, some joint models have been proposed to model these two tasks simultaneously. Early feature-based joint models (Yu and Lam, 2010; Miwa and Sasaki, 2014) rely on complicated feature engineering to build interaction between entities and relations. More recently, neural joint models have attracted increasing research interest and have demonstrated promising performance on joint entity and relation extraction.

In existing neural joint models, there are mainly two ways to build the interaction between NER and RE: parameter sharing and joint decoding. In parameter sharing methods (Zeng et al., 2018; Bekoulis et al., 2018a; Dixit and Al-Onaizan, 2019), NER model and RE model are built on top of a shared encoding layer to achieve joint learning. However, approaches based on parameter sharing implicitly rather than explicitly model the inter-task interaction, leading to insufficient excavation of the inherent association between the two tasks. Moreover, these two tasks focus on different contextual information (Zhong and Chen, 2021; Ye et al., 2022), but methods of sharing representations cannot provide task-specific features with enough specificity for the two tasks. In terms of error propagation, parameter sharing methods alleviate the error propagation between tasks, but to a limited extent, because these models still perform pipelined decoding.

[*]Corresponding author.

Proceedings of the 21st China National Conference on Computational Linguistics, pages 848-860, Nanchang, China, October 14 - 16, 2022.
(c) Technical Committee on Computational Linguistics, Chinese Information Processing Society of China

Another family of approaches adopt unified tagging framework in the form of sequences (Zheng et al., 2017), tables (Zhang et al., 2017; Ren et al., 2021), or graphs (Fu et al., 2019; Xue et al., 2021) to integrate the information of entities and relations as a whole and perform joint decoding to extract relational triples. Although these methods enhance the inter-task interaction, the specificity of task features is not well considered since the entities and relations still share contextual representations in essence. Moreover, all these joint decoding methods require complex joint decoding algorithms, and it is challenging to balance the accuracy of joint decoding and the abundance of task-specific features.

Accordingly, the main challenge of joint entity and relation extraction is to construct proper interaction between NER and RE while ensuring the specificity of task-specific features. Wang and Lu (2020) adopt two types of representations to generate task-specific representations, sequence representations for NER and table representations for RE, separately. These two types of representations interact with each other to model inter-task interaction. Yan et al. (2021) perform neuron partition in an autoregressive manner to generate task-specific features jointly in order to build inter-task interaction. They combine the task-specific features and global features as the final input to the task modules. Inspired by Yan et al. (2021)'s work, we adopt the task modules they used that model each relation separately with tables (Miwa and Sasaki, 2014), and we propose a simple but effective feature encoding approach for joint entity and relation extraction, achieving excellent results while being less computationally intensive. We will detail the differences and our advantages in Section 3.5.

In this work, we propose a **M**ulti-**G**ate **E**ncoder (**MGE**) that control the flow of feature information based on gating mechanism, so as to filter out undesired information and retain desired information. MGE has two types of gates: task gates and interaction gates. Task gates are used to generate task-specific features, and interaction gates control how much information flows out to guide the opposite task. The output of interaction gate is combined with the opposite task-specific features to generate the input of corresponding task module, resulting in a bidirectional interaction between NER and RE while maintaining sufficient specificity of task-specific features.

The main contributions of this work are summarized below:

1. A multi-gate encoder for joint entity and relation extraction is proposed, which effectively promotes interaction between NER and RE while ensuring the specificity of task features. Experimental results show that our method establishes the new state-of-the-art on three standard benchmarks, namely ACE04, ACE05, and SciERC.
2. We conduct extensive analyses to investigate the superiority of our model and validate the effectiveness of each component of our model.
3. The effect of relation information on entity recognition is examined. Our additional experiments suggest that relation information contributes to predicting entities, which helps clarify the controversy on the effect of relation signals.

## 2 Related Work

The task of extracting relational triples from plain text can be decomposed into two sub-task: Named Entity Recognition and Relation Extraction. The two tasks can be performed in a pipelined manner (Chan and Roth, 2011; Gormley et al., 2015; Zhong and Chen, 2021; Ye et al., 2022) or in a joint manner (Miwa and Sasaki, 2014; Zheng et al., 2017; Wang and Lu, 2020; Yan et al., 2021).

Traditional pipelined methods (Zhou et al., 2005; Chan and Roth, 2011; Gormley et al., 2015) firstly train a model to extract entities and then train another model to classify the relation type between subject and object for each entity pair. Recent pipelined approaches (Zhong and Chen, 2021; Ye et al., 2022) still follow this pattern and adopt marker-based span representations to learn different contextual representations between entities and relations, and between entity pairs, which sheds some light on the importance of feature specificity. Although Zhong and Chen (2021) and Ye et al. (2022) achieve better performance than previous pipelined methods and some joint methods, they still run the risk of error propagation and do not adequately account for interactions between tasks. To ease these issues, some joint models that extract entities and relations jointly has been proposed.

Joint entity and relation extraction is a typical multi-task scenario, and how to handle the interaction between tasks is a frequently discussed topic. Early joint models (Yu and Lam, 2010; Miwa and Sasaki, 2014) rely on feature engineering to build task interaction. More recently, many neural joint models have been proposed and show promising performance. Miwa and Bansal (2016) builds a sequence tagging model for NER and a dependency tree model for RE separately on top of a shared LSTM layer and performs joint learning, achieving task interaction through parameter sharing. Zeng et al. (2018) uses sequence-to-sequence learning framework with copy mechanism to jointly extract entities and relations. Bekoulis et al. (2018b) builds a CRF layer for NER and a sigmoid layer for RE on a shared LSTM layer. Eberts and Ulges (2020) proposes a span-based joint model for entity and relation extraction. They performs span classification and span filtering to extract entity spans and then performs relation classification based on the contextual span representations from BERT (Devlin et al., 2019) encoder. All these approaches construct the interaction between NER and RE through parameter sharing. Another class of methods adopts joint decoding to fuse the two tasks together. Li and Ji (2014) uses structured perceptron with beam search to extract entities and relations simultaneously. Wang et al. (2018) proposes a transition system to convert the joint task into a directed graph. Wang et al. (2020b) introduces a novel handshaking tagging scheme to formulate joint extraction as a token pair linking problem. Zhang et al. (2017) and Ren et al. (2021) convert the task into a table-filling problem.

In addition to building interaction between tasks, another important issue is the specificity of task features. As recent studies (Zhong and Chen, 2021; Ye et al., 2022) have shown, generating specific contextual features for different tasks can achieve better results on the overall task than sharing input features. Zhong and Chen (2021) and Ye et al. (2022) both use a pre-trained language model (e.g., BERT) for NER and another for RE to obtain different contextual representations for specific task. However, fine-tuning distinct pre-trained encoders for the two task separately is computationally expensive. In our work, we adopts gating mechanism to balance the flow of feature information, taking into account both the interaction between tasks and the specificity of task features.

## 3 Method

In this section, we first formally define the problem of joint entity and relation extraction and then detail the structure of our model. Finally, we discuss how our model differs from the approach we follow and explain why our method performs better.

### 3.1 Problem Definition

The problem of joint entity and relation extraction can be decomposed into two sub-tasks: NER and RE. Let $\mathcal{E}$ denotes the set of predefined entity types and $\mathcal{R}$ denotes the set of predefined relation types. Given a sentence containing $N$ words, $X = \{x_1, x_2, \ldots, x_N\}$, the goal of NER is to extract an entity type $e_{ij} \in \mathcal{E}$ for each span $s_{ij} \in S$ that starts with $x_i$ and ends with $x_j$, where $S$ is the set of all the possible spans in $X$. For RE, the goal is to extract a relation type $r_{i_1 i_2} \in \mathcal{R}$ for each span pair whose start words are $x_{i_1}$ and $x_{i_2}$ respectively. Combining the results of NER and RE, we get the final output of this problem $Y_r = \{(e_{i_1 j_1}, r_{i_1 i_2}, e_{i_2 j_2})\}$, where $e_{i_1 j_1}, e_{i_2 j_2} \in \mathcal{E}, r_{i_1 i_2} \in \mathcal{R}$.

### 3.2 Multi-Gate Encoder

We adopt BERT (Devlin et al., 2019) to encode the contextual information of input sentences. As shown in Figure 1, our proposed MGE employs four gates to control the flow of feature information based on gating mechanism. The two task gates are designed to generate task-specific features for NER and RE, while the two interaction gates aim to generate interaction features that have a positive effect on the opposite task. The task-specific features and interaction features are combined to form the input of task modules, carrying out bidirectional task interaction through feature exchange.

Let $H_b \in \mathbb{R}^{N \times d}$ denotes the contextual feature matrix of sentence $X$ extracted by BERT encoder, where $d$ is the hidden size of BERT layer. In order to preliminarily build the specificity between entity recognition features and relation extraction features, we generate candidate entity features $H_e^c$ and candidate relation

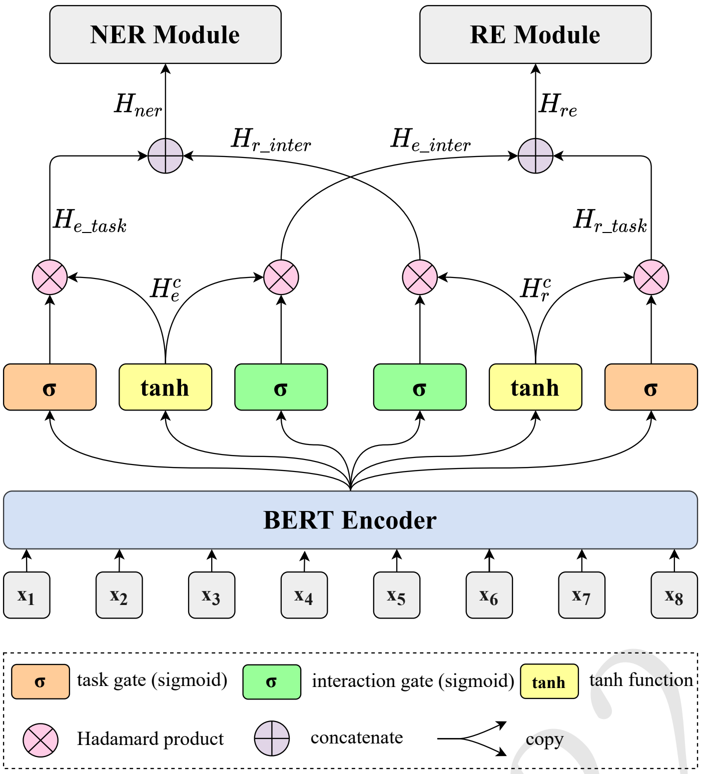

Figure 1: The architecture of our proposed MGE. There are two types of gates in the encoder: task gates and interaction gates. $H_e^c$ and $H_r^c$ denote candidate entity features and candidate relation features respectively. $H_{e\_task}$ and $H_{r\_task}$ denote task-specific features generated by task gates. $H_{e\_inter}$ and $H_{r\_inter}$ denote interaction features generated by interaction gates to guide the opposite task. $H_{ner}$ and $H_{re}$ are the final input features to NER module and RE module.

features $H_r^c$ based on BERT output representations as follows:

$$\begin{aligned} H_e^c &= \tanh\left(H_b W_e + b_e\right) \\ H_r^c &= \tanh\left(H_b W_r + b_r\right), \end{aligned} \tag{1}$$

where $W_{(\cdot)} \in \mathbb{R}^{d\times h}$ and $b_{(\cdot)} \in \mathbb{R}^h$ denote trainable weights and bias and $h$ is the hidden size in MGE. $\tanh(\cdot)$ means $tanh$ activation function. The candidate features will be input to the task gates and interaction gates of corresponding task for further feature filtering to generate task-specific features and interaction features.

The task gates decide what information in the candidate features is contributing to the corresponding specific task, which is implemented by a sigmoid layer. The sigmoid layer produces values in the range of zero to one, indicating how much information is to be transmitted. A value of zero means no information is allowed to pass, whereas a value of one means all the information is allowed to pass. We calculate entity task gate $G_{e\_task}$ and relation task gate $G_{r\_task}$ as below:

$$\begin{aligned} G_{e\_task} &= \sigma\left(H_b W_{e\_task} + b_{e\_task}\right) \\ G_{r\_task} &= \sigma\left(H_b W_{r\_task} + b_{r\_task}\right), \end{aligned} \tag{2}$$

where $\sigma(\cdot)$ represents sigmoid activation function. $W_{(\cdot)} \in \mathbb{R}^{d\times h}$ and $b_{(\cdot)} \in \mathbb{R}^h$ denote weights and bias. The entity task gate $G_{e\_task}$ and relation task gate $G_{r\_task}$ work independently and are specialized in filtering information useful for specific task in candidate features to obtain task-specific features for entity recognition and relation extraction respectively. We calculate the Hadamard (element-wise) product between task gates and candidate features to generate task-specific features for NER and RE:

$$\begin{aligned} H_{e\_task} &= G_{e\_task} \odot H_e^c \\ H_{r\_task} &= G_{r\_task} \odot H_r^c, \end{aligned} \tag{3}$$

where $\odot$ denotes Hadamard product operation. $H_{e\_task}$ and $H_{r\_task}$ represent entity task-specific features and relation task-specific features respectively.

Similarly, the interaction gates decide what information in entity candidate features $H_e^c$ is helpful for guiding relation extraction and what information in $H_r^c$ is helpful for guiding entity recognition. This is also implemented through sigmoid activation function:

$$\begin{aligned} G_{e\_inter} &= \sigma\left(H_b W_{e\_inter} + b_{e\_inter}\right) \\ G_{r\_inter} &= \sigma\left(H_b W_{r\_inter} + b_{r\_inter}\right), \end{aligned} \tag{4}$$

where $G_{e\_inter}$ denotes entity interaction gate and $G_{r\_inter}$ denotes relation interaction gate. $W_{(\cdot)} \in \mathbb{R}^{d \times h}$ and $b_{(\cdot)} \in \mathbb{R}^h$ denote weights and bias. These two interaction gates are then applied to candidate features to generate interaction features:

$$\begin{aligned} H_{e\_inter} &= G_{e\_inter} \odot H_e^c \\ H_{r\_inter} &= G_{r\_inter} \odot H_r^c, \end{aligned} \tag{5}$$

where $H_{e\_inter}$ denotes entity interaction features used to guide RE and $H_{r\_inter}$ denotes relation interaction features used to guide NER.

Finally, we perform feature exchange based on the task-specific features and interaction features to achieve bidirectional interaction between NER and RE. Specifically, we concatenate entity task-specific features $H_{e\_task}$ and relation interaction features $H_{r\_inter}$, and relation task-specific features $H_{r\_task}$ is concatenated with entity interaction features $H_{e\_inter}$:

$$\begin{aligned} H_{ner} &= H_{e\_task} \oplus H_{r\_inter} \\ H_{re} &= H_{r\_task} \oplus H_{e\_inter}, \end{aligned} \tag{6}$$

where $\oplus$ means concatenation operation. $H_{ner} \in \mathbb{R}^{N \times 2h}$ and $H_{re} \in \mathbb{R}^{N \times 2h}$ are the final features to be input to NER and RE task modules respectively. Through exchanging features that are designed to guide the opposite task and combining task-specific features, $H_{ner}$ and $H_{re}$ balance the task interaction and feature specificity of NER and RE.

### 3.3 Table-filling Modules

Following Yan et al. (2021), we adopt table-filling framework to extract entities and relations, which treats both NER and RE as table filling problems. For NER, the goal is to predict all the entity spans and corresponding entity types. Specifically, we construct a $N \times N$ type-specific table for each entity type $k \in \mathcal{E}$, whose element at row $i$ and column $j$ represents the probability of span $s_{ij} \in S$ belonging to type $k$. We firstly concatenate the representations of every two tokens based on $H_{ner}$ and connect a fully-connected layer to reduce the hidden size. Then we employ layer normalization (Ba et al., 2016) and ELU activation (Clevert et al., 2015) to obtain table representations of spans. Formally, for span $s_{ij}$ that starts with $x_i$ and ends with $x_j$, we compute the table representation $T_{ner}^{i,j} \in \mathbb{R}^h$ as follows:

$$T_{ner}^{i,j} = \mathrm{ELU}(\mathrm{LayerNorm}([H_{ner}^i; H_{ner}^j]W_e^h + b_e^h)), \tag{7}$$

where $H_{ner}^i \in \mathbb{R}^{2h}$ and $H_{ner}^j \in \mathbb{R}^{2h}$ denote the vectors corresponding to words $x_i$ and $x_j$ in entity features $H_{ner} \in \mathbb{R}^{N \times 2h}$ that containing both entity task-specific information and relation interaction information. $W_e^h \in \mathbb{R}^{4h \times h}$ and $b_e^h \in \mathbb{R}^h$ are trainable parameters. To predict the probability of span $s_{ij}$ belonging to entity type $k$, we project the hidden size to $|\mathcal{E}|$ with a fully-connected layer followed by a sigmoid activation function:

$$p(e_{ij} = k) = \sigma(T_{ner}^{i,j} W_e^{tag} + b_e^{tag}), \forall k \in \mathcal{E}, \tag{8}$$

where $W_e^{tag} \in \mathbb{R}^{h \times |\mathcal{E}|}$ and $b_e^{tag} \in \mathbb{R}^{|\mathcal{E}|}$ are trainable parameters and $|\mathcal{E}|$ represents the number of predefined entity types.

The goal of RE table-filling module is to predict the start word of each entity and classify the relations between them. The structure of RE module is formally analogous to the NER module. Similar to NER, we construct a $N \times N$ type-specific table for each relation type $l \in \mathcal{R}$. For the table corresponding to relation $l$, the element at row $i$ and column $j$ represents the probability that the $i$-th word $x_i$ and the $j$-th word $x_j$ in a sentence are respectively the start words of subject entity and object entity of relation type $l$. For $x_i$ and $x_j$, we compute the table representations $T_{re}^{i,j} \in \mathbb{R}^h$ as follows:

$$T_{re}^{i,j} = \mathrm{ELU}(\mathrm{LayerNorm}([H_{re}^i; H_{re}^j]W_r^h + b_r^h)), \tag{9}$$

where $H_{re}^i \in \mathbb{R}^{2h}$ and $H_{re}^j \in \mathbb{R}^{2h}$ denote the vectors corresponding to words $x_i$ and $x_j$ in features $H_{re} \in \mathbb{R}^{N \times 2h}$ that containing both relation task-specific information and entity interaction information. $W_r^h \in \mathbb{R}^{4h \times h}$ and $b_r^h \in \mathbb{R}^h$ are trainable parameters. The probability that $x_i$ and $x_j$ are the start words of the subject and object of relation type $l$ is calculated as follows:

$$p(r_{ij} = l) = \sigma(T_{re}^{i,j} W_r^{tag} + b_r^{tag}), \forall l \in \mathcal{R}, \tag{10}$$

where $W_r^{tag} \in \mathbb{R}^{h \times |\mathcal{R}|}$ and $b_r^{tag} \in \mathbb{R}^{|\mathcal{R}|}$ are trainable parameters and $|\mathcal{R}|$ represents the number of predefined relation types. We obtain the prediction results of NER module and RE module under the following conditions:

$$p(e_{i_1 j_1} = k_1) \geq 0.5;\ p(r_{i_1 i_2} = l) \geq 0.5;\ p(e_{i_2 j_2} = k_2) \geq 0.5 \tag{11}$$

where $k_1, k_2 \in \mathcal{E}, l \in \mathcal{R}$. For a fair comparison, the hyper-parameter threshold is set to be 0.5 without further fine-tuning as in previous works.

Combining the prediction results of NER and RE task modules, we can get the final relational triples in a given sentence:

$$Y_r = \{(e_{i_1 j_1}, r_{i_1 i_2}, e_{i_2 j_2})\}, e_{i_1 j_1}, e_{i_2 j_2} \in \mathcal{E}, r_{i_1 i_2} \in \mathcal{R}, \tag{12}$$

where $e_{i_1 j_1}$ and $e_{i_2 j_2}$ are entity spans predicted by NER task module, and $r_{i_1 i_2}$ denotes the relation between head-only entities predicted by RE task module.

### 3.4 Loss Function

During training, we adopt binary cross entropy loss for both NER and RE task modules. Given a sentence containing $N$ words, we compute the NER loss and RE loss as follows:

$$\begin{aligned} \mathcal{L}_{\text{NER}} &= -\sum_{i=1}^{N}\sum_{j=i}^{N}\sum_{k \in \mathcal{E}} \hat{p}(e_{ij} = k)\log p(e_{ij} = k) + (1 - \hat{p}(e_{ij} = k))\log(1 - p(e_{ij} = k)) \\ \mathcal{L}_{\text{RE}} &= -\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{l \in \mathcal{R}} \hat{p}(r_{ij} = l)\log p(r_{ij} = l) + (1 - \hat{p}(r_{ij} = l))\log(1 - p(r_{ij} = l)), \end{aligned} \tag{13}$$

where $\hat{p}(e_{ij} = k)$ and $\hat{p}(r_{ij} = l)$ represent ground truth labels. $p(e_{ij} = k)$ and $p(r_{ij} = l)$ are the probability predicted by NER and RE modules. The final training goal is to minimize the sum of these two losses:

$$\mathcal{L} = \mathcal{L}_{\text{NER}} + \mathcal{L}_{\text{RE}}. \tag{14}$$

### 3.5 Differences from PFN

Our method differs from PFN (Yan et al., 2021) in the following ways: (1) We generate interaction features using distinct interaction gates, which are independent of the process of generating task-specific features. (2) All feature operations in MGE are performed in a non-autoregressive manner, i.e., all tokens in the sentence are processed in a single pass, resulting in increased efficiency. As a result, our method is simpler while still ensuring proper NER-RE interaction. Furthermore, as demonstrated in Section 4, our model outperforms PFN on three public datasets and achieves faster inference speed while employing the same task modules and pre-trained encoders.

## 4 Experiments

### 4.1 Dataset

We evaluate our model on three popular English relation extraction datasets: ACE05 (Walker et al., 2006), ACE04 (Doddington et al., 2004) and SciERC (Luan et al., 2018). The ACE05 and ACE04 datasets are collected from various domains, such as news articles and online forums. Following Luan et al. (2019), we split ACE04 into 5 folds and ACE05 into 10051 sentences for training, 2424 sentences for validation, and 2050 sentences for test [0]. And we follow Yan et al. (2021) to construct the development set of ACE04 with 15% of the training set.

| **Dataset** | $\lvert\mathcal{E}\rvert$ | $\lvert\mathcal{R}\rvert$ | **#Entities** | **#Relations** | **#Sentences** | | |
|---|---|---|---|---|---|---|---|
| | | | | | Train | Dev | Test |
| ACE05 | 7 | 6 | 38,287 | 7,070 | 10,051 | 2,424 | 2,050 |
| ACE04 | 7 | 6 | 22,708 | 4,084 | 8, 683 (5-fold) | | |
| SciERC | 6 | 7 | 8,094 | 4,684 | 1,861 | 275 | 551 |

Table 1: Statistics of datasets. $|\mathcal{E}|$ and $|\mathcal{R}|$ are numbers of entity and relation types.

The SciERC dataset is collected from 500 AI paper abstracts, and includes annotations for scientific entities, their relations, and coreference clusters. It consists six predefined scientific entity types and seven predefined relation types. In our experiments, we only use the annotation information of entities and relations. We download the processed dataset from the project website [1] of Luan et al. (2018), including 1861 sentences for training, 275 sentences for validation and 551 sentences for test. Table 1 shows the statistics of ACE04, ACE05 and SciERC datasets.

### 4.2 Evaluation

Following standard evaluation protocol, we use micro F1 score as an evaluation for both NER and RE. For NER task, an entity is considered as correct if its boundary and type are both predicted correctly. For RE task, a relational triple is correct only if its relation type and the boundaries and types of entities are correct.

### 4.3 Implementation Details

For fair comparison, we use *albert-xxlarge-v1* (Lan et al., 2020) as the base encoder for ACE04 and ACE05. And for SciERC, we use *scibert-scivocab-uncased* (Beltagy et al., 2019) as the base encoder. Regarding the use of cross-sentence context (Luan et al., 2019; Luoma and Pyysalo, 2020), that is, to extend each sentence by its context for better contextual representations, we don't adopt this experimental setting considering the fairness of experimental comparisons. Zhong and Chen (2021) extend each sentence to a fixed context window size of 300 words for entity model and 100 words for relation model. Ye et al. (2022) set the context window size to be 512 words for entity model and 256 / 384 words for relation model. Although cross-sentence context may further enhance the performance of entity recognition and relation extraction, if the research focus is not on the cross-sentence context, the different cross-sentence context lengths will greatly affect the experimental results, making it difficult to conduct fair comparisons. All our experiments are carried out in single-sentence setting and we compare with the experimental results of other baselines under the single-sentence setting.

Our model is implemented with PyTorch and we train our models with Adam optimizer of a linear scheduler with a warmup ratio of 0.1. For all the experiments, the learning rate and training epoch are set to be 2e-5 and 100 respectively. We set the batch size to be 4 for SciERC and 16 for ACE04 and ACE05. Following previous work (Yan et al., 2021), the max length of input sentence is set to be 128. All the

[0] We process the datasets with scripts provided by Luan et al. (2019): https://github.com/luanyi/DyGIE/tree/master/preprocessing.

[1] http://nlp.cs.washington.edu/sciIE/

| Model | Encoder | ACE05 | | ACE04 | | SciERC | |
|---|---|---|---|---|---|---|---|
| | | NER | RE | NER | RE | NER | RE |
| SPTree (Miwa and Bansal, 2016) | LSTM | 83.4 | 55.6 | 81.8 | 48.4 | - | - |
| Katiyar and Cardie (2017) | LSTM | 82.6 | 53.6 | 79.6 | 45.7 | - | - |
| Multi-turn QA (Li et al., 2019) | BERT | 84.8 | 60.2 | 83.6 | 49.4 | - | - |
| Table-Sequence (Wang and Lu, 2020) | ALBERT | 89.5 | 64.3 | 88.6 | 59.6 | - | - |
| SPE (Wang et al., 2020a) | SciBERT | - | - | - | - | 68.0 | 34.6 |
| PURE (Zhong and Chen, 2021) | ALBERT | **89.7** | 65.6 | 88.8 | 60.2 | - | - |
| | SciBERT | - | - | - | - | 66.6 | 35.6 |
| PFN (Yan et al., 2021) | ALBERT | 89.0 | 66.8 | **89.3** | 62.5 | - | - |
| | SciBERT | - | - | - | - | 66.8 | 38.4 |
| MGE (Ours) | ALBERT | **89.7** | **68.2** | **89.3** | **63.8** | - | - |
| | SciBERT | - | - | - | - | **68.4** | **39.4** |

Table 2: Overall F1 scores on the test set of ACE04, ACE05, and SciERC. Results of PURE are reported in single-sentence setting for fair comparison.

models are trained with a single NVIDIA Titan RTX GPU. We select the model with the best average F1 score of NER and RE on the development set, and report the average F1 of 5 runs on the test set.

## 4.4 Baselines

We compare our model with the following baselines: (1) **BiLSTM** (Miwa and Bansal, 2016; Katiyar and Cardie, 2017): these models perform NER and RE based on shared Bi-directional LSTMs. Miwa and Bansal (2016) treats entity recognition as a sequence tagging task and represents the relations between entities in dependency tree. Katiyar and Cardie (2017) formulates both entity recognition and relation detection as sequence tagging tasks. (2) **Multi-turn QA** (Li et al., 2019): it converts the task into a multi-turn question answering task: each entity type and relation type has its corresponding pre-designed question template, and entities and relations are extracted by answering template questions with standard machine reading comprehension (MRC) (Seo et al., 2018) framework. (3) **Table-Sequence** (Wang and Lu, 2020): this work uses a sequence encoder and a table encoder to learn task-specific representations for NER and RE separately, and models task interaction through combining these two types of representations. (4) **SPE** (Wang et al., 2020a): this method proposes a span encoder and span pair encoder to add intra-span and inter-span information to the pre-trained model for entity and relation extraction task. (5) **PURE** (Zhong and Chen, 2021): this work builds two independent encoders for NER and RE separately and performs entity relation extraction in a pipelined fashion. PURE experimentally validates the importance of learning different contextual representations for entities and relations separately. (6) **PFN** (Yan et al., 2021): this work proposes a partition filter network to generate task-specific features and shared features of the two tasks, and then combining global features to extract entities and relations with table-filling framework.

Among these baselines, the two BiLSTM based methods build task interaction through parameter sharing, Multi-turn QA is a paradigm shift based method, PURE is a pipelined method, and Table-Sequence, SPE and PFN are methods based on multiple representations interaction.

## 4.5 Main Results

Table 2 reports the results of our approach MGE compared with other baselines on ACE05, ACE04 and SciERC. As is shown, MGE achieves the best results in terms of F1 score against all the comparison baselines. For NER, MGE achieves similar performance to PURE (Zhong and Chen, 2021) on ACE05 but surpasses PURE by an absolute entity F1 of +0.5%, +1.8% on ACE04 and SciERC. And for RE, our method obtains a substantially +2.6%, +3.6%, +3.8% absolute relation F1 improvement over PURE on ACE05, ACE04, and SciERC respectively. This demonstrates the superiority of the bidirectional task

| Model | SciERC | | ACE05 | |
|---|---|---|---|---|
| | RE (F1) | Speed (sent/s) | RE (F1) | Speed (sent/s) |
| PFN | 38.4 | 342.2 | 66.8 / 60.8† | 34.2 / 387.2† |
| MGE (Ours) | **39.4** | **479.2** | **68.2 / 62.0†** | **36.0 / 567.6†** |

Table 3: We compare our MGE model with PFN model in both relation F1 and inference speed. We use $scibert - scivocab - uncased$ for SciERC and $albert - xxlarge - v1$ / $bert - base - cased$ for ACE05. † marks the inference speed on ACE05 when using $bert - base - cased$ encoder. The speed is measured on a single NVIDIA Titan V GPU with a batch size of 32.

interaction in our model compared to the unidirectional interaction in PURE.

In comparison to the previous state-of-the-art model PFN (Yan et al., 2021), we can see that our method achieves a similar entity F1 to PFN on ACE04, but an absolute relation F1 improvement of +1.3%. This suggests that, given the same NER performance, our method can obtain a better RE performance, implying that the entity knowledge in our method more effectively leads the RE task. Furthermore, on ACE05, MGE surpasses PFN by an absolute F1 improvement of +0.7% and +1.4% in NER and RE, respectively. On SciERC, we get a 1.6% higher entity F1 and a 1.0% higher relation F1 compared to PFN. Note that we use the same pre-trained encoders and task modules as PFN, and these improvements demonstrate the effectiveness of our proposed multi-gate encoder.

### 4.6 Inference Speed

As described in Section 3.5, our method employs a non-autoregressive way for feature encoding, which is simpler and faster than the autoregressive approach in PFN. In order to experimentally compare the model efficiency, we conduct experiments to evaluate these two models' inference speed on the test set of ACE05 and SciERC datasets. We perform inference experiments on a single NVIDIA Titan V GPU with a batch size of 32.

Table 3 shows the relation F1 scores and the inference speed of PFN and MGE. We use $scibert - scivocab - uncased$ encoder for SciERC and $albert - xxlarge - v1$ / $bert - base - cased$ (Devlin et al., 2019) encoder for ACE05. As is shown, with the same pre-trained model, our method obtains +1.0% improvement in relation F1 score with +40% speedup on the test set of SciERC. On ACE05, our model achieves a relation F1 improvement of +1.4% compared to PFN, but only slightly accelerates the inference speed (34.2 vs 36.0) when using $albert - xxlarge - v1$ pre-trained model. This is because $albert - xxlarge - v1$ contains 223M parameters, which is much larger than the 110M parameters in $scibert - scivocab - uncased$ and $bert - base - cased$, and most of the computational cost of the model is concentrated in the pre-trained model part. As a result, the speedup provided by MGE does not appear to be significant. Therefore, we also evaluate the inference speed on ACE05 using $bert - base - cased$. As Table 3 shows, our model achieves +47% speedup and an absolute relation F1 improvement of +1.2% on ACE05 when using $bert - base - cased$. This clearly demonstrates that our proposed MGE can improve the performance of joint entity and relation extraction while accelerating the model inference speed.

## 5 Analysis

In this section, we conduct ablation study on ACE05, ACE04 and SciERC to investigate how each component of MGE affects the final performance, where we apply $albert - xxlarge - v1$ encoder for ACE05 and ACE04, $scibert - scivocab - uncased$ encoder for SciERC. Specifically, we ablate the task gate or interaction gate to verify their effectiveness.

### 5.1 Effect of Task Gates.

We remove task gates from the complete MGE structure to explore whether they can generate effective task-specific features. As shown in Table 4, when we remove the entity task gate, the entity F1 scores on the ACE04 and SciERC datasets decrease by 0.5% and 0.2%, respectively. And when we remove the

| Encoder | | | | | ACE05 | | ACE04 | | SciERC | |
|---|---|---|---|---|---|---|---|---|---|---|
| B | $G_{e\_task}$ | $G_{r\_task}$ | $G_{e\_inter}$ | $G_{r\_inter}$ | NER | RE | NER | RE | NER | RE |
| √ | √ | √ | √ | √ | 89.7 | **68.2** | **89.3** | **63.8** | 68.4 | **39.4** |
| √ | - | √ | √ | √ | 89.7 | 67.4 | 88.8 | 62.2 | 68.2 | 37.5 |
| √ | √ | - | √ | √ | 89.9 | 67.8 | 88.8 | 62.6 | 68.0 | 39.1 |
| √ | √ | √ | - | √ | 89.4 | 67.4 | 89.1 | 63.0 | **68.5** | 38.9 |
| √ | √ | √ | √ | - | **90.0** | 66.6 | 89.2 | 63.6 | 68.2 | 38.7 |
| √ | √ | √ | - | - | **90.0** | 66.1 | 88.4 | 62.8 | 67.9 | 37.8 |

Table 4: F1 scores of ablation study on ACE05, ACE04 and SciERC. B denotes BERT encoder. $G_{e\_task}$, $G_{r\_task}$, $G_{e\_inter}$ and $G_{r\_inter}$ means entity task gate, relation task gate, entity interaction gate and relation interaction gate.

relation task gate, the relation F1 scores on ACE05, ACE04 and SciERC datasets decrease by 0.4%, 1.2% and 0.3%, respectively. This indicates that task gates can effectively generate task-specific features to improve the performance of NER and RE.

### 5.2 Effect of Interaction Gates.

We also investigate the effect of the MGE entity interaction gate and relation interaction gate on task interaction. As there is no entity interaction gate, it is similar to weakening the guidance of entity information on the relation extraction task when compared to the unaffected MGE model. After deleting the entity interaction gate, the relation F1 scores on the ACE05, ACE04, and SciERC datasets decrease by 0.8%, 0.8%, and 0.5%, respectively, as shown in Table 4. In MGE, this highlights the effectiveness of the entity interaction gate.

Although it is widely accepted that entity information is necessary for relation extraction, previous research on the impact of relation information on entity recognition has been mixed. Zhong and Chen (2021) claims that relation information has no significant improvement on entity model. However, Yan et al. (2021) discover that relation signals have a significant impact on entity prediction. Our research also sheds light on this contentious issue. In MGE, the guidance of relation information on entity recognition is cut off when the relation interaction gate is ablate. The entity F1 scores decrease on ACE04 and SciERC but increase on ACE05 when the relation interaction gate is removed. Our experimental results match the experimental analysis of Yan et al. (2021). They conclude that relation information is helpful for predicting entities that appear in relational triples, but not for entities outside relational triples. According to Yan et al. (2021), there are fewer entities belonging to relational triples in ACE05, compared with ACE04 and SciERC. Consequently, the relation information is comparatively less helpful for entity recognition in ACE05 but has a positive effect on entity recognition in ACE04 and SciERC. To sum up, the relation interaction gate can effectively generate interaction features to facilitate the recognition of entities within triples.

Moreover, when we remove both the entity interaction gate and the relation interaction gate, the relation F1 scores on ACE05, ACE04 and SciERC datasets decrease by 2.1%, 1.0% and 1.6%, respectively. This shows the effectiveness of interaction gates in MGE for task interaction in joint entity relation extraction.

### 5.3 Bidirectional Interaction Vs Unidirectional Interaction.

From Table 4, we also observe that employing only an entity interaction gate or only a relation interaction gate in the encoder performs worse than adopting these two gates simultaneously. This means that the two tasks of entity recognition and relation extraction are mutually reinforcing, and bidirectional interaction between NER and RE is more effective than unidirectional interaction.

## 6 Conclusion

In this paper, we propose a multi-gate encoder for joint entity and relation extraction. Our model adopts gate mechanism to build bidirectional task interaction while ensuring the specificity of task features by controlling the flow of feature information. Experimental results on three standard benchmarks show that our model achieves state-of-the-art F1 scores for both NER and RE. We conduct extensive analyses on three datasets to investigate the superiority of our model and validate the effectiveness of each component of our model. Furthermore, our ablation study suggests that relation information contributes to entity recognition, which helps to clarify the controversy on the effect of relation information.

## Acknowledgements

This work was supported by the National Defense Science and Technology Key Laboratory Fund Project of the Chinese Academy of Sciences: Space Science and Application of Big Data Knowledge Graph Construction and Intelligent Application Research and Manned Space Engineering Project: Research on Technology and Method of Engineering Big Data Knowledge Mining.

## References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450.*

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, Brussels, Belgium, October-November. Association for Computational Linguistics.

Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. 2018b. Joint entity recognition and relation extraction as a multi-head selection problem. *CoRR*, abs/1804.07847.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China, November. Association for Computational Linguistics.

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

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. 2015. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289.*

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

Kalpit Dixit and Yaser Al-Onaizan. 2019. Span-level model for relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5308–5314, Florence, Italy, July. Association for Computational Linguistics.

George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. The automatic content extraction (ACE) program – tasks, data, and evaluation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal, May. European Language Resources Association (ELRA).

Markus Eberts and Adrian Ulges. 2020. Span-based joint entity and relation extraction with transformer pre-training. In *ECAI 2020*, pages 2006–2013. IOS Press.

Radu Florian, Abe Ittycheriah, Hongyan Jing, and Tong Zhang. 2003. Named entity recognition through classifier combination. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 168–171.

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

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, Lisbon, Portugal, September. Association for Computational Linguistics.

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

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite bert for self-supervised learning of language representations. In *iclr*.

Qi Li and Heng Ji. 2014. Incremental joint extraction of entity mentions and relations. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 402–412, Baltimore, Maryland, June. Association for Computational Linguistics.

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

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Twenty-ninth AAAI conference on artificial intelligence*.

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

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

Jouni Luoma and Sampo Pyysalo. 2020. Exploring cross-sentence contexts for named entity recognition with BERT. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 904–914, Barcelona, Spain (Online), December. International Committee on Computational Linguistics.

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

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 (EMNLP)*, pages 1858–1869, Doha, Qatar, October. Association for Computational Linguistics.

Feiliang Ren, Longhui Zhang, Shujuan Yin, Xiaofeng Zhao, Shilei Liu, Bochao Li, and Yaduo Liu. 2021. A novel global feature-oriented relational triple extraction model based on table filling. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2646–2656, Online and Punta Cana, Dominican Republic, November. Association for Computational Linguistics.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2018. Bidirectional Attention Flow for Machine Comprehension, June. Number: arXiv:1611.01603 arXiv:1611.01603 [cs].

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus. *Linguistic Data Consortium, Philadelphia*, 57:45.

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)*, pages 1706–1721, Online, November. Association for Computational Linguistics.

Shaolei Wang, Yue Zhang, Wanxiang Che, and Ting Liu. 2018. Joint extraction of entities and relations based on a novel graph scheme. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, IJCAI'18, page 4461–4467. AAAI Press.

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

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

Fuzhao Xue, Aixin Sun, Hao Zhang, and Eng Siong Chng. 2021. Gdpnet: Refining latent multi-view graph for relation extraction. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI*, pages 2–9.

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

Deming Ye, Yankai Lin, Peng Li, and Maosong Sun. 2022. Pack together: Entity and relation extraction with levitated marker. In *Proceedings of ACL 2022*.

Xiaofeng Yu and Wai Lam. 2010. Jointly identifying entities and extracting relations in encyclopedia text via a graphical model approach. In *Coling 2010: Posters*, pages 1399–1407, Beijing, China, August. Coling 2010 Organizing Committee.

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

Xiangrong Zeng, Daojian Zeng, Shizhu He, Kang Liu, and Jun Zhao. 2018. Extracting relational facts by an end-to-end neural model with copy mechanism. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 506–514, Melbourne, Australia, July. Association for Computational Linguistics.

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

Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. 2017. Joint extraction of entities and relations based on a novel tagging scheme. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1227–1236, Vancouver, Canada, July. Association for Computational Linguistics.

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

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, Ann Arbor, Michigan, June. Association for Computational Linguistics.