RFBFN: A Relation-First Blank Filling Network for Joint Relational Triple Extraction

Zhe Li[1], **Luoyi Fu**[1], **Haisong Zhang**[3], **Chenghu Zhou**[2], **Xinbing Wang**[1]
[1]School of Electronic Information and Electrical Engineering,
Shanghai Jiao Tong University, Shanghai, China
[2]Chinese Academy of Sciences, Beijing, China
[3]Tencent AI Lab, Shenzhen, China
{lizhe2016,yiluofu,xwang8}@sjtu.edu.cn
zhouch@lreis.ac.cn, hansonzhang@tencent.com

Abstract

Joint relational triple extraction from unstructured text is an important task in information extraction. However, most existing works either ignore the semantic information of relations or predict subjects and objects sequentially. To address the issues, we introduce a new blank filling paradigm for the task, and propose a relation-first blank filling network (RFBFN). Specifically, we first detect potential relations maintained in the text to aid the following entity pair extraction. Then, we transform relations into relation templates with blanks which contain the fine-grained semantic representation of the relations. Finally, corresponding subjects and objects are extracted simultaneously by filling the blanks. We evaluate the proposed model on public benchmark datasets. Experimental results show our model outperforms current state-of-the-art methods. The source code of our work is available at: https://github.com/lizhe2016/RFBFN.

1 Introduction

Extracting pairs of entities with semantic relations from unstructured texts is essential in knowledge graph construction. Given a text, the aim of this task is to detect triples, i.e., in the form of *(subject, relation, object)* or *(s, r, o)*. Traditional pipeline methods (Chan and Roth, 2011; Lin et al., 2016) first extract entity mentions and then perform relation classification for each entity pair. However, they suffer from error propagation and ignore the interaction between the two tasks.

Different from the pipeline methods, joint learning methods (Yu et al., 2020; Zeng et al., 2020; Zheng et al., 2021) aim to extract entities and relations simultaneously in an end-to-end way, which achieve promising performance. They tend to decompose the task into several subtasks and solve the problem through a multi-task learning framework (Miwa and Bansal, 2016; Wei et al., 2020; Zheng et al., 2021).

Model	Relation Semantics	Relation-First Prediction	Simultaneous Subject-Object Extraction
Multi-Turn QA (Li et al., 2019)	Yes	No	No
PRGC (Zheng et al., 2021)	No	Yes	No
RFBFN (Ours)	Yes	Yes	Yes

Table 1: Comparison of our RFBFN and previous methods.

Although previous works have achieved great success, the semantic information of relations is still underutilized. Most models (Miwa and Bansal, 2016; Zeng et al., 2018; Zhong and Chen, 2021) treat the relation extraction as a classification task which only replace the relation with a meaningless class ID. To better capture the semantic information, machine reading comprehension (MRC) models (Li et al., 2019; Zhao et al., 2020; Goswami et al., 2020) are proposed to address the extraction task. Li et al. (2019) and Zhao et al. (2020) transform the task into a multi-turn question answering problem. The subjects are detected first by answering entity-specific questions. Then, relation-specific questions are generated to extract objects. However, they predict subjects and objects sequentially and separately, and thus question answering is required to perform for multiple turns.

More recently, the relation-first methods have shown promising performance in relational triple extraction (Zheng et al., 2021; Ma et al., 2021), which benefit from the fact that relations are usually triggered by the context rather than entities. For example, the *"creator"* relation will be directly detected from descriptions such as *"was created by"*. By predicting relations first, irrelevant relations are filtered out, which mitigates negative effects caused by useless relations and avoids the data imbalance issue. However, the subject-object

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Student Research Workshop, pages 10 - 20
May 22-27, 2022 ©2022 Association for Computational Linguistics

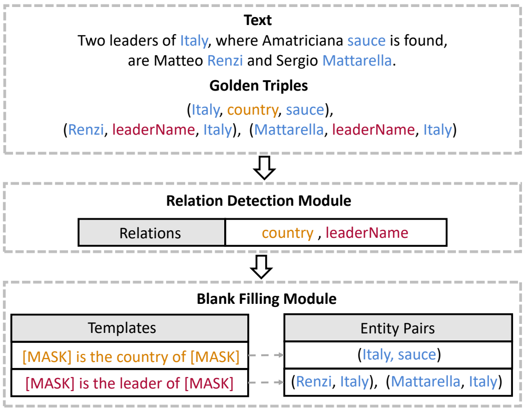
Text
Two leaders of Italy, where Amatriciana sauce is found, are Matteo Renzi and Sergio Mattarella.
Golden Triples
(Italy, country, sauce),
(Renzi, leaderName, Italy), (Mattarella, leaderName, Italy)
Relation Detection Module
Relations
country , leaderName
Blank Filling Module
Templates
[MASK] is the country of [MASK]
[MASK] is the leader of [MASK]
Entity Pairs
(Italy, sauce)
(Renzi, Italy), (Mattarella, Italy)

Figure 1: An illustration of the relational triple extraction in the proposed RFBFN. The relation templates contain blanks for entity extraction.

alignment mechanism is needed to align subjects and objects to form valid triples in these works. We review and compare previous methods in Table 1.

We propose an end-to-end relation-first framework for joint relational triple extraction, which can not only capture the semantics of relations, but also extract subjects and objects simultaneously. We formalize the task as a relation-first blank filling problem, inspired by the cloze task (Taylor, 1953). Our RFBFN includes a relation detection module and a blank filling module. For the relation detection module, we first obtain a subset of most relevant relations and filter out irrelevant ones. For the blank filling module, we transform relations to relation templates which contain significant semantics of relations. As shown in Figure 1, the model needs to fill the blanks in the templates like "[MASK] is the country of [MASK]" and "[MASK] is the leader of [MASK]" with the corresponding subjects and objects. Thus, entity pairs in the text which have the corresponding relations will be extracted by filling the blanks. Notably, our model detects subjects and objects simultaneously in a non-autoregressive decoder without aligning them. Besides, entities are allowed to be assigned with different relations, which naturally tackles the overlapping cases. Experiments on public datasets demonstrate that our proposed method outperforms the state-of-the-art methods. The main contributions of this paper are as follows:

- We propose a novel end-to-end relation-first blank filling network for relational triple extraction, which first detects relations, and then extracts subjects and objects simultaneously in a non-autoregressive transformer decoder.
- We tackle the entity pair extraction from a novel perspective which transforms the task to a blank filling problem. This paradigm allows the model to encode the prior knowledge of the relations in the templates and make use of semantic information of the relations.
- Extensive experiments on two public datasets show that the proposed framework achieves state-of-the-art results, especially for complex scenarios of overlapping triples. Further ablation studies and analyses confirm the effectiveness of our model.

2 Related Work

Early works (Zelenko et al., 2003; Chan and Roth, 2011; Lin et al., 2016) treat the extraction as a pipeline of two separate tasks: an entity model first identifies entities and then a relation model extracts the relations between the entity mentions. However, these methods ignore the correlation between the two steps and suffer from the error propagation issue. To overcome these shortcomings, joint models (Lin et al., 2020; Wang and Lu, 2020) are proposed, which can extract entities and relations simultaneously.

Traditional joint methods (Yu and Lam, 2010; Li and Ji, 2014; Miwa and Sasaki, 2014; Ren et al., 2017) are feature-based and heavily rely on feature engineering, which require intensive manual efforts. To reduce manual work, recent studies have investigated neural network models, which include sequence tagging methods (Zheng et al., 2017; Dai et al., 2019; Yu et al., 2020), sequence-to-sequence methods (Zeng et al., 2018, 2020) and table-filling methods (Gupta et al., 2016; Wang et al., 2021).

Although above models make great progress, they still only treat the relation type as a meaningless class ID or a trainable embedding (Yuan et al., 2020; Zheng et al., 2021) which is not enough to capture the fine-grained semantic information of a relation. Current works cast the task into a question answering problem with machine reading models. Goswami et al. (2020) perform unsupervised relation extraction without a fine-tuned extractive head. However, they only extract objects from the given contexts and subjects. To joint extract entities and relations, Li et al. (2019); Zhao et al. (2020) first predict subjects from the context by answering entity questions. Then, the extracted subjects are inserted to the slots to generate the relation ques-

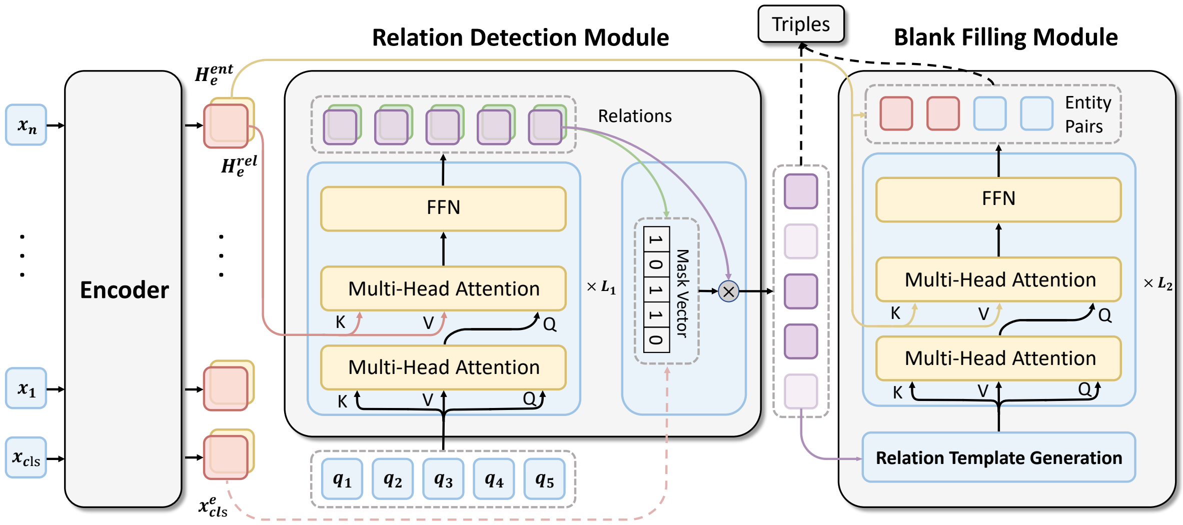

Figure 2: The overall architecture of RFBFN. Given a sentence X, RFBFN first predicts a subset of candidate relations in the relation detection module. Then for each candidate relation, corresponding entity pairs are extracted by filling the blanks of the transformed relation templates in the blank filling module. $q_1, q_2, ..., q_5$ are learnable embeddings to predict relations. L_1 and L_2 are the numbers of the decoder blocks.

tions and then objects can be extracted. Although the well developed machine reading comprehension models can be exploited, they extract subjects and objects sequentially and need multiple turns.

In this paper, we propose a joint relation-first blank filling network to extract triples. Different from previous works, we transform relations to specific relation templates to make use of semantic information of the relations. Moreover, we extract subjects and objects at the same time in a non-autoregressive decoder without aligning them.

3 Method

3.1 Overview

For relational triple extraction task, the input is a sentence $X = (x_1, x_2, ..., x_n)$, which comprises n tokens of the sentence with another special `[CLS]` token x_{cls}. Let $\mathcal{R}$ be the set of predefined relation types. The task is to predict all possible triples as $T(X) = (e_i, r_{ij}, e_j)$, where e_i, e_j are sequences of tokens denoting the subject and object respectively, and $r_{ij} \in \mathcal{R}$ is the relation that holds between e_i and e_j.

Figure 2 shows an overview architecture of the proposed RFBFN. It consists of three main parts: *Span-Level Encoder*, *Relation Detection Module* and *Blank Filling Module*. First, the encoder preprocesses the source text and extracts the span representations. Then the relation detection module predicts potential relations and filters out irrelevant ones. Finally, the blank filling module takes a set of relation templates as input and predicts the corresponding entity pairs. We model relation extraction as a blank filling task, which can not only capture the semantics of a relation, but also extract subjects and objects simultaneously.

3.2 Span-Level Encoder

The goal of this component is to obtain the contextualized representation of each span in a sentence. We utilize BERT (Devlin et al., 2019) as the feature encoder due to its effectiveness in representation learning. Let $S = (s_1, s_2, ..., s_{n_s})$ be all possible spans in X. Given a span $s_i \in S$, the span representation $\mathrm{h}_\mathrm{i}^\mathrm{e}$ is defined as:

$$\mathrm{h}_\mathrm{i}^\mathrm{e} = [\mathbf{x}_{\mathrm{START(i)}}^\mathrm{e}; \mathbf{x}_{\mathrm{END(i)}}^\mathrm{e}; \phi(\mathbf{x}_\mathrm{i})], \qquad (1)$$

where $\mathbf{x}_{\mathrm{START(i)}}^\mathrm{e}$ and $\mathbf{x}_{\mathrm{START(i)}}^\mathrm{e}$ are the context-aware representations of the boundary tokens. $\phi(\mathbf{x}_i)$ represents the feature vector denoting the span length (Wadden et al., 2019; Zhong and Chen, 2021). Unlike the token-level models, overlapping spans can be detected because each span is independent of others. The output of the encoder is the representation of spans, and is denoted as $\mathbf{H}^\mathrm{e} \in \mathbb{R}^{\mathrm{n_s} \times \mathrm{d}}$, where n_s is the number of spans and d is embedding dimension.

Then $\mathbf{H}^\mathrm{e}$ is fed into two separate *Feed-Forward Networks (FFN)* to generate the features for the *Relation Detection Module* and the *Blank Filling*

Module respectively:

$$\mathbf{H}_{\mathrm{e}}^{\mathrm{rel}} = \mathbf{W}_{rel}\mathbf{H}^{e} + \mathbf{b}_{rel}, \\ \mathbf{H}_{\mathrm{e}}^{\mathrm{ent}} = \mathbf{W}_{ent}\mathbf{H}^{e} + \mathbf{b}_{ent}, \tag{2}$$

where $\mathbf{W}_{rel}, \mathbf{W}_{ent} \in \mathbb{R}^{d\times d}$ are trainable weights and $\mathbf{b}_{rel}, \mathbf{b}_{ent} \in \mathbb{R}^{d}$ are trainable biases.

3.3 Relation Detection Module

Different from previous works (Yuan et al., 2020; Wei et al., 2020) which redundantly perform entity extraction to every relation, we first predict a subset of candidate relations in a sentence, then entities only need to be extracted based on these target ones. This module first predicts potential relations with a non-autoregressive decoder, then irrelevant ones are excluded with a binary classifier.

Potential Relation Extractor We predict the relations with the transformer-based non-autoregressive decoder (Vaswani et al., 2017), as shown in Figure 2. The input of the decoder is initialized by n_q learnable embeddings $\mathbf{Q} \in \mathbb{R}^{\mathrm{n_q}\times\mathrm{d}}$, where n_q is set to be the maximum number of relations in a sentence. Different from the prior token-level cross-attention, we exploit the span representation $\mathbf{H}_{\mathrm{e}}^{\mathrm{rel}}$ as part of the input here. Given the output embedding $\mathbf{H}^{\mathrm{r}} \in \mathbb{R}^{\mathrm{n_q}\times\mathrm{d}}$, the predicted relation type is obtained by:

$$\mathbf{p}_{\mathrm{i}}^{\mathrm{r}} = \mathrm{Softmax}(\mathbf{W}_{\mathrm{r}}\mathbf{h}_{\mathrm{i}}^{\mathrm{r}} + \mathbf{b}_{\mathrm{r}}), \tag{3}$$

where $\mathbf{W}_{\mathrm{r}} \in \mathbb{R}^{|\mathcal{R}|\times\mathrm{d}}$, $\mathbf{b}_{\mathrm{r}} \in \mathbb{R}^{|\mathcal{R}|}$ are learnable parameters and $|\mathcal{R}|$ is the total number of relation types. We adopt the bipartite matching loss (Sui et al., 2020) in the training process, which is invariant to any permutation of predictions.

Candidate Relation Judgement After predicting a subset of potential relations, we filter out irrelevant ones to generate relation templates effectively. Given the output representation matrix $\mathbf{H}^{\mathrm{r}}$ of the non-autoregressive decoder and the embedding of `[CLS]`, this component predicts a boolean mask vector $\mathbf{M}$ from a binary classifier to guide the candidate relation set:

$$\mathbf{M} = \sigma(\mathbf{W}_{\mathrm{s}}[\mathbf{H}^{\mathrm{r}}; \mathbf{x}_{\mathrm{cls}}^{\mathrm{e}}] + \mathbf{b}_{\mathrm{s}}), \tag{4}$$

where $\mathbf{W}_s$ is the trainable weight, $\mathbf{b}_s$ is the bias and σ is the sigmoid activation function. The higher the value, the higher the confidence level that the relation contains in a sentence, and vice versa. In this step, for each sentence, we filter out useless relations and predict a subset $\mathcal{R}_i \in \mathcal{R}$ to discard most of the negative samples. If the text contains the j-th relation type, it will be fed into blank filling module to aid entity pair recognition.

3.4 Blank Filling Module

We propose a new blank filling paradigm for entity pair extraction, i.e., the extraction of entity pairs is transformed to the task of identifying answer spans from the context to fill the blanks. We transform each candidate relation type to a template with blanks (denoted as `[MASK]` here), which are then filled with the participating subjects and objects. In other words, if the context contains the corresponding entity pairs of the relation, entity spans will be extracted by filling the blanks.

Relation Template Generation Each relation type is associated with a type-specific template. A relation template is generated manually by combing the semantic information and two blanks as shown in Figure 1. For example, the relation "`leaderName`" corresponds to the template like "`[MASK] is the leader of [MASK]`". The relation template encodes the semantic information for the relation which is important for relational triple extraction. Formally, the input relation template can be denoted as:

$$T_r = (m_1^r, t_1^r, t_2^r, ..., t_{nt}^r, m_2^r), \tag{5}$$

where m_1^r denotes the blank for the subject, m_2^r for the object and $t_1^r, t_2^r, ..., t_{nt}^r$ are the relation tokens of the relation r. Each relation template is copied k times and then concatenated with the special `[SEP]` token, where k is larger than the typical triple number of the relation. Therefore, multiple entity pairs with the same relation can be extracted in one pass.

Entity Pair Extractor Given the relation template and the span representation $\bar{\mathbf{H}} = [\mathbf{H}_{\mathrm{e}}^{\mathrm{ent}}; \mathbf{x}_{\mathrm{cls}}^{\mathrm{e}}]$, the goal of this component is to extract corresponding entity pairs. We use a non-autoregressive span-level transformer decoder as our entity pair extractor, which is similar to the relation extractor. In each transformer layer, the multi-head self-attention is to model the association between blanks and relation semantics, and the multi-head cross-attention is to fuse the information of the spans. After the decoder, blanks are embedded into $\mathbf{H}_{\mathrm{r}}^{\mathrm{blk}} \in \mathbb{R}^{2\mathrm{k}\times\mathrm{d}}$.

Next, the decoder copies subjects and objects from possible spans in the source sentence as the

Dataset	#Relations	#Sentences			Details of Test Set				
		Train	Valid	Test	Normal	EPO	SEO	$N = 1$	$N > 1$
NYT*	24	56195	4999	5000	3266	978	1297	3244	1756
WebNLG*	171	5019	500	703	246	26	457	266	437
NYT	24	56196	5000	5000	3071	1168	1273	3089	1911
WebNLG	216	5019	500	703	239	6	448	256	447

Table 2: Statistics of the datasets in experiments, where N is the number of triples in a sentence. EPO and SEO refer to entity pair overlapping and single entity overlapping respectively (Zeng et al., 2018). Note that a sentence can belong to both EPO and SEO patterns.

predictions of the blanks in parallel. To handle the instances without corresponding entities, we set the answer as the `[CLS]` token. We calculate the span representations for each blank as:

$$\mathbf{h}^{\mathrm{b}}_{\mathrm{i,r}} = \tanh(\mathbf{W}^{1}_{\mathrm{b}}\bar{\mathbf{H}} + \mathbf{W}^{2}_{\mathrm{b}}\mathbf{h}^{\mathrm{blk}}_{\mathrm{i,r}} + \mathbf{b}_{\mathrm{b}}), \qquad (6)$$

where $\mathbf{W}^{1}_{\mathrm{b}}, \mathbf{W}^{2}_{\mathrm{b}} \in \mathbb{R}^{\mathrm{d}\times\mathrm{d}}$ are the trainable weights and $\mathbf{b}_{\mathrm{b}} \in \mathbb{R}^{\mathrm{d}}$ is the trainable bias.

Finally, we apply softmax to obtain the probability distribution and select the span with the highest probability as the predicted entity:

$$\mathbf{p}^{\mathrm{b}}_{\mathrm{i,r}} = \mathrm{Softmax}(\mathbf{u}^{\mathrm{T}}_{\mathrm{b}} \cdot \mathbf{h}^{\mathrm{b}}_{\mathrm{i,r}}), \qquad (7)$$

where $\mathbf{u}_b \in \mathbb{R}^d$ is the learnable parameter. We use the span-based method to predict entity pairs, so entities with multiple tokens can be extracted simultaneously without the pointer network or the sequence labeling scheme.

3.5 Joint Training

There are totally two tasks in our model: relation detection and entity pair extraction. During optimization, we train the model jointly in a multi-task manner and share the parameters of the encoder. To predict entity pairs, we sort them according to their order in the text, and adopt cross-entropy loss as the loss function for entity pair extraction:

$$\mathcal{L}_{ent} = -\sum_{r=1}^{n_d}\sum_{i=1}^{2k} \log \mathbf{p}^{\mathrm{b}}_{\mathrm{i,r}}(\mathrm{y}^{\mathrm{b}}_{\mathrm{i,r}}), \qquad (8)$$

where $\mathrm{y}^{\mathrm{b}}_{\mathrm{i,r}}$ is the ground truth entity span for relation r and n_d is the detected relation number. However, for relation detection, there exists no suitable way to sort the relations, thus we adopt bipartite matching loss (Sui et al., 2020) which does not penalize small order shift. To find an optimal matching between the ground truth relations and predicted relations, we search for a permutation strategy π^* with the lowest cost:

$$\pi^* = \operatorname*{argmin}_{\pi\in\Pi(\mathrm{n_q})}\Big(-\sum_{\mathrm{i}=1}^{\mathrm{n_q}} \mathrm{I}(\mathrm{y}^{\mathrm{r}}_{\mathrm{i}}) \cdot \mathbf{p}^{\mathrm{r}}_{\pi(\mathrm{i})}(\mathrm{y}^{\mathrm{r}}_{\mathrm{i}})\Big), \qquad (9)$$

where $\Pi(n_q)$ is the space of all permutation strategies, $\mathrm{y}^{\mathrm{r}}_{\mathrm{i}}$ is the ground truth relation. $I(y^r_i)$ is a switching function: if $\mathrm{y}^{\mathrm{r}}_{\mathrm{i}} \neq \varnothing$, $\mathrm{I}(\mathrm{y}^{\mathrm{r}}_{\mathrm{i}}) = 1$, otherwise 0. We define the loss for relation detection as:

$$\mathcal{L}_{rel} = -\sum_{\mathrm{i}=1}^{\mathrm{n_q}} \log \mathbf{p}^{\mathrm{r}}_{\pi^*(\mathrm{i})}(\mathrm{y}^{\mathrm{r}}_{\mathrm{i}}) \qquad (10)$$

The total loss is the sum of two parts:

$$\mathcal{L} = \lambda\mathcal{L}_{ent} + (1-\lambda)\mathcal{L}_{rel}, \qquad (11)$$

where $\lambda \in \mathbb{R}$ is the parameter controlling the trade-off between the two objectives. During the training phase, the model learns to minimize $\mathcal{L}$ and optimizes the parameters jointly.

4 Experiments

4.1 Experimental Settings

Datasets We evaluate our approach on two benchmark datasets: NYT24 (Riedel et al., 2010) and WebNLG (Gardent et al., 2017). Both of them have two different versions. NYT* and WebNLG* annotate the last word of entities, while NYT and WebNLG annotate the whole entity span. We use the datasets released by (Zheng et al., 2021), in which the statistics of the datasets are shown in Table 2. To further study the capability of RFBFN in extracting overlapping and multiple relations, we also split the test set by overlapping patterns (Zeng et al., 2018) and triple numbers.

Baselines and Evaluation Metrics We compare our model with eleven strong baseline models including the state-of-the-art model GRTE$_{BERT}$ (Ren et al., 2021). The experimental results of the baseline models are from the original papers.

Model	NYT*			WebNLG*			NYT			WebNLG		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
NovalTagging (Zheng et al., 2017)	-	-	-	-	-	-	32.8	30.6	31.7	52.5	19.3	28.3
CopyRE (Zeng et al., 2018)	61.0	56.6	58.7	37.7	36.4	37.1	-	-	-	-	-	-
MutiHead (Bekoulis et al., 2018)	-	-	-	-	-	-	60.7	58.6	59.6	57.5	54.1	55.7
GraphRel (Fu et al., 2019)	63.9	60.0	61.9	44.7	41.1	42.9	-	-	-	-	-	-
ETL-span (Yu et al., 2020)	84.9	72.3	78.1	84.0	91.5	87.6	85.5	71.7	78.0	84.3	82.0	83.1
CasRel$_{BERT}$ (Wei et al., 2020)	89.7	89.5	89.6	93.4	90.1	91.8	-	-	-	-	-	-
TPLinker$_{BERT}$ (Wang et al., 2020)	91.3	92.5	91.9	91.8	92.0	91.9	91.4	92.6	92.0	88.9	84.5	86.7
SPN$_{BERT}$ (Sui et al., 2020)	93.3	91.7	92.5	93.1	93.6	93.4	92.5	92.2	92.3	-	-	-
PRGC$_{Random}$ (Zheng et al., 2021)	89.6	82.3	85.8	90.6	88.5	89.5	87.8	83.8	85.8	82.5	79.2	80.8
PRGC$_{BERT}$ (Zheng et al., 2021)	93.3	91.9	92.6	**94.0**	92.1	93.0	93.5	91.9	92.7	89.9	87.2	88.5
GRTE$_{BERT}$ (Ren et al., 2021)	92.9	93.1	93.0	93.7	**94.2**	93.9	93.4	93.5	93.4	**92.3**	87.9	90.0
RFBFN$_{Random}$	88.6	86.8	87.7	90.4	90.8	90.6	87.9	86.1	87.0	83.1	82.1	82.6
RFBFN$_{BERT}$	**93.4**	**93.2**	**93.3**	93.9	94.1	**94.0**	**93.7**	**93.6**	**93.6**	91.5	**89.4**	**90.4**

Table 3: Comparison of the proposed RFBFN method with the prior works. **Bold** marks the highest score. The subscript *Random* refers to a model with randomly initialized parameters.

In our experiments, to keep in line with previous works (Sui et al., 2020; Zheng et al., 2021; Ren et al., 2021), an extracted triple is regarded as correct only if it is an extract match with ground truth, which means the last word of entities in NYT* and WebNLG* or the whole entity span in NYT and WebNLG of both subject and object and the relation are all correct. The standard micro precision, recall, and F1 score are used to evaluate the results.

Implementation Details For fair comparison, we use the BERT-Base-Cased English model[1] as our embedding layer. We train our model with AdamW optimizer with batch size of 8 for 100 epochs. We set the learning rate $1e-5$ for the pre-trained parameters, $5e-5$ for cross-attention and $7e-5$ for others. The spans are up to 8 words and $\lambda = 0.5$ for loss. The duplicate number k of relation templates on NYT*, NYT, WebNLG* and WebNLG is set to 6, 8, 3 and 3 respectively. The learnable embedding number n_q is set to 15/12 in NYT(NYT*)/WebNLG(WebNLG*).

4.2 Main Results

The results of our model against other baseline methods are shown in Table 3. Our RFBFN model outperforms them in respect of almost all evaluation metrics even if compared with the recent strongest baseline (Ren et al., 2021). We also implement RFBFN$_{Random}$ where all parameters are randomly initialized. Especially, RFBFN$_{Random}$ improves 1.9% F1 on NYT*, 1.1% F1 on WebNLG*, 1.2% F1 on NYT and 1.8% F1 on WebNLG over PRGC$_{Random}$. The performance of RFBFN$_{Random}$ demonstrates that our framework still achieves better results than others which do not take BERT as the pre-trained language model.

Our RFBFN outperforms the most competitive GRTE$_{BERT}$ model in four F1 scores. There are two main reasons behind this. First, the relation detection module greatly reduces irrelevant relations compared to GRTE$_{BERT}$ which generates a table feature for each relation. In other words, filtering negative relations provides additional benefits compared to the models which perform entity extraction under every relation. Second, introduction of semantic information of the relations is significant for relational triple extraction. However, GRTE$_{BERT}$ only assigns trainable weights for the relations, which can not fully explore the semantic information of the relations. Moreover, our model detects subjects and objects simultaneously in the non-autoregressive decoder. By contrast, PRGC$_{BERT}$ is a relation-first model, which extracts subjects and objects in two separate sequence tagging operations and needs to check the corresponding score in a global matrix for subject-object alignment. We find that detects subjects and objects simultaneously can achieve better results.

4.3 Detailed Results on Complex Scenarios

Following previous works (Sui et al., 2020; Zheng et al., 2021; Ren et al., 2021), we conduct further experiments on NYT* and WebNLG* to verify

[1] Available at https://huggingface.co/bert-base-cased.

Model	NYT*								WebNLG*							
	Normal	SEO	EPO	$N=1$	$N=2$	$N=3$	$N=4$	$N\geq 5$	Normal	SEO	EPO	$N=1$	$N=2$	$N=3$	$N=4$	$N\geq 5$
CasRel	87.3	91.4	92.0	88.2	90.3	91.9	94.2	83.7	89.4	92.2	94.7	89.3	90.8	94.2	92.4	90.9
TPLinker	90.1	93.4	94.0	90.0	92.8	93.1	96.1	90.0	87.9	92.5	95.3	88.0	90.1	94.6	93.3	91.6
SPN	90.8	94.0	94.1	90.9	93.4	94.2	95.5	90.6	89.5	94.1	90.8	89.5	91.3	96.4	94.7	93.8
PRGC	91.0	94.0	94.5	91.1	93.0	93.5	95.5	93.0	90.4	93.6	95.9	89.9	91.6	95.0	94.8	92.8
GRTE	91.1	94.4	95.0	90.8	93.7	94.4	96.2	93.4	90.6	94.5	96.0	90.6	92.5	96.5	**95.5**	94.4
RFBFN	**91.2**	**95.2**	**95.6**	**91.4**	**93.8**	**94.8**	**96.4**	**93.9**	**91.0**	**94.6**	**96.5**	**90.8**	**92.6**	**96.6**	94.7	**94.5**

Table 4: F1 score on sentences with different overlapping patterns and different triple numbers. N is the number of triples in a sentence.

	Subtask	Prec.	Rec.	F1
NYT*	Potential Relation Extractor	96.8	96.0	96.4
	Candidate Relation Judgement	97.7	95.4	96.5
	Entity Pair Extractor	95.0	94.8	94.9
	Combination of Above All	93.4	93.2	93.3
WebNLG*	Potential Relation Extractor	95.8	95.9	95.9
	Candidate Relation Judgement	96.9	94.9	95.9
	Entity Pair Extractor	96.5	96.7	96.6
	Combination of Above All	93.9	94.1	94.0

Table 5: Results of different subtasks on NYT* and WebNLG* datasets. Relation performance after *Potential Relation Extractor* and *Candidate Relation Judgement*. Entity performance after *Entity Pair Extractor*.

the capability of our model in handling different overlapping patterns and sentences with different numbers of triples. As shown in Table 4, we can see that RFBFN achieves the best results on all three overlapping patterns of both datasets. Besides, the performance of our model is better than others almost for all numbers of triples. In general, these two further experiments adequately show the advantages of our model in complex scenarios.

4.4 Results on Different Subtasks

To further verify the results of the subtasks, we present more detailed evaluations on NYT* and WebNLG* datasets which show the performance after each component of our model in Table 5. After the *Candidate Relation Judgement* component, we get higher precision in relation detection to reduce negative relations and ensure most detected relations are correct. In the *Entity Pair Extractor* component, golden relation templates are taken as input, which showcases the upper bound result that our model can achieve for relational triple extraction. The result shows the proposed blank filling module outperforms existing models by a large margin (up to 2.7%). This indicates that our method is able to capture the sufficient semantic information of relations which helps to extract entities.

Model	Prec.	Rec.	F1
RFBFN	**93.9**	94.1	**94.0**
– Relation Detection Module	81.7	89.0	85.2
– Candidate Relation Judgement	92.9	**94.3**	93.6
– Relation Template Generation	93.0	93.2	93.1
– Non-Autoregressive Entity Pair Extractor	88.8	88.2	88.5
– Joint Training	92.4	92.6	92.5

Table 6: Ablation study on WebNLG* dataset.

For NYT*, we find that identifying relations is somehow easier than identifying entities. In contrast to NYT*, for WebNLG*, it is more challenging to identify the relations than entities, as the performance of the entity pair extractor is much higher than the overall performance. We attribute the difference to the different numbers of relations in two datasets (24 in NYT* and 171 in WebNLG*), which make identification of relations much harder in WebNLG*.

5 Analysis

5.1 Ablation Study

We conduct ablation experiments to evaluate the contributions of some main components in RFBFN. We remove one component at a time to obtain its impact on the experimental results, which is summarized in Table 6.

(1) – *Relation Detection Module* denotes that the model removes the *Relation Detection Module* from RFBFN, and uses all relations to extract entity pairs. It is not possible to enumerate all relations in WebNLG* (171 in all), and thus we randomly add 30% negative ones. As shown in Table 6, the performance significantly decreases without relation detection. It is because that redundant relations cause negative influence on entity pair extractor. Meanwhile, with the increase of relation number,

Texts	Ground Truth	Embeddings	Relation Templates
Acta Mathematica **Hungarica** is the publisher of Springer Science + Business **Media**, founded by Julius **Springer**.	(Hungarica, publisher, Media) (Springer, founder, Media)	(Springer, publisher, Media) ✖ (Springer, founder, Media)	(Hungarica, publisher, Media) (Springer, founder, Media)
Buzz **Aldrin** is a national of the United **States** whose leader is Joe **Biden**. He was born in Glen Ridge, Essex County, New **Jersey**.	(Jersey, birthPlace, Aldrin) (States, nationality, Aldrin) (Biden, leaderName, States) (Jersey, isPartOf, Jersey)	(Jersey, birthPlace, Aldrin) (States, nationality, Aldrin) (Biden, leaderName, Jersey) ✖ (Jersey, isPartOf, Jersey)	(Jersey, birthPlace, Aldrin) (States, nationality, Aldrin) (Biden, leaderName, States) (Jersey, isPartOf, Jersey)

Figure 3: Case study for ablation study of *–Relation Template Generation*. Examples are from WebNLG* dataset. The correct entities are in **bold**, the correct relations are colored and the red cross marks bad cases.

it results in a heavy computational burden.

(2) – *Candidate Relation Judgement* denotes that the model ablates the *Candidate Relation Judgement* component from RFBFN, which ignores the impact of negative relations. We note the performance decreases in the result, which indicates that this component contributes to reducing the noise brought by unrelated relations. In other words, filtering out irrelevant relations is helpful for relational triple extraction.

(3) – *Relation Template Generation* denotes that the model replaces relation templates with trainable embeddings. As shown in the results, the performance drops significantly. Through the case study in Figure 3, we observe that if the relation is only represented by a trainable embedding, the model cannot understand the underlying semantics of a relation and predicts wrong entity pairs. Although it has the ability to detect right entities, it ignores their relation. However, our relation template can capture fine-grained semantic information of the relation, which is helpful for extracting entities. We argue that the explicit semantic representation of a relation plays an important role for relational triple extraction which is ignored in most previous works.

(4) – *Non-Autoregressive Entity Pair Extractor* denotes that the decoder replaces the unmasked self-attention with the casual mask and the entity pair extractor starts with a detected relation. In this way, subjects and objects are generated sequentially. The results in Table 6 reveal that predicting subjects and objects simultaneously in our non-autoregressive decoder is reasonable.

(5) – *Joint Training* denotes that the relation detection module and the blank filling module are trained separately without parameter sharing. As shown in Table 6, joint learning framework brings a remarkable improvement (1.5%) in F1 score, which demonstrates that our potential relation extractor and entity pair extractor actually work in a mutually beneficial way.

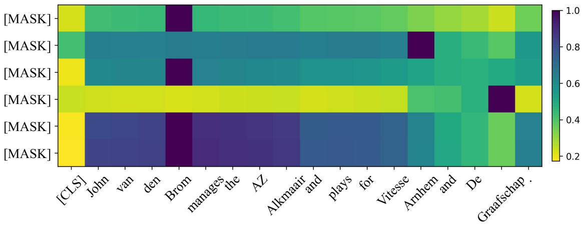

Figure 4: An illustration on how different blanks attend to the words in the text. The attention score is averaged over all attention heads in the last layer. The darker color denotes the higher score.

5.2 Visualization

In order to validate that our model is able to fill the blanks with related entities in the sentence, we visualize the cross-attention score of the blank filling module in Figure 4. The source sentence contains two triples, i.e. *(Brom, club, Arnhem)*, *(Brom, club, Graafschap)* and the input relation of the entity pair extractor is `club`. As shown in Figure 4, through span-level cross-attention, different blanks can attend to corresponding entities with the specific relation. In the entity pair extractor, subjects and objects with the same relation can be extracted simultaneously rather than sequentially. Besides, the extracting order is determined with the sorting scheme, thus we do not extract repetitive entity pairs. The visualization demonstrates the validity of our model.

6 Conclusion

In this paper, we design a novel blank filling paradigm for relational triple extraction, and present a relation-first blank filling network. We transform relations into relation templates with blanks to fill which can capture important semantic information of the relations. Meanwhile, subjects and objects are extracted simultaneously by filling the blanks in the non-autoregressive decoder. To the best of our knowledge, we are the first to cast relational triple extraction as a blank filling problem, which may motivate new ideas and inspire future research directions. The experiment results on public datasets show that our model achieves state-of-the-art performance.

Acknowledgements

This work is supported by National Key R&D Program of China (No.2018YFB2100302), NSF China (No. 42050105, 62020106005, 62061146002, 61960206002, 61822206, 61832013, 61829201), 2021 Tencent AI Lab RhinoBird Focused Research Program (No: JR202132), and the Program of Shanghai Academic/Technology Research Leader under Grant No. 18XD1401800.

References

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

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

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. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):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, Minneapolis, Minnesota. Association for Computational Linguistics.

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

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

Ankur Goswami, Akshata Bhat, Hadar Ohana, and Theodoros Rekatsinas. 2020. Unsupervised relation extraction from language models using constrained cloze completion. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1263–1276, Online. Association for Computational Linguistics.

Pankaj Gupta, Hinrich Schütze, and Bernt Andrassy. 2016. Table filling multi-task recurrent neural network for joint entity and relation extraction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2537–2547, Osaka, Japan. The COLING 2016 Organizing Committee.

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

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

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

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

Lianbo Ma, Huimin Ren, and Xiliang Zhang. 2021. Effective cascade dual-decoder model for joint entity and relation extraction. *arXiv preprint arXiv:2106.14163*.

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

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. 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. Association for Computational Linguistics.

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. WWW '17, page 1015–1024, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases*, pages 148–163, Berlin, Heidelberg. Springer Berlin Heidelberg.

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

Wilson L Taylor. 1953. "cloze procedure": A new tool for measuring readability. *Journalism quarterly*, 30(4):415–433.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

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 (EMNLP-IJCNLP)*, pages 5784–5789, Hong Kong, China. Association for Computational Linguistics.

Jue Wang and Wei Lu. 2020. Two are better than one: Joint entity and relation extraction with table-sequence encoders. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1706–1721, Online. Association for Computational Linguistics.

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

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

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

Bowen Yu, Zhenyu Zhang, Xiaobo Shu, Tingwen Liu, Yubin Wang, Bin Wang, and Sujian Li. 2020. Joint extraction of entities and relations based on a novel decomposition strategy. In *ECAI*, pages 2282–2289. IOS Press.

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. Coling 2010 Organizing Committee.

Yue Yuan, Xiaofei Zhou, Shirui Pan, Qiannan Zhu, Zeliang Song, and Li Guo. 2020. A relation-specific attention network for joint entity and relation extraction. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 4054–4060. International Joint Conferences on Artificial Intelligence Organization. Main track.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. *Journal of machine learning research*, 3(Feb):1083–1106.

Daojian Zeng, Haoran Zhang, and Qianying Liu. 2020. Copymtl: Copy mechanism for joint extraction of entities and relations with multi-task learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9507–9514.

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

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

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

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

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