Fuzzy Systems and Data Mining IX
A.J. Tallón-Ballesteros and R. Beltrán-Barba (Eds.)

doi:10.3233/FAIA231047

RSFnet: A Relation Semantic Fusion-Based Entity Relation Extraction Method

Ruiran Ding [a], Ling Lu [a,1], Qiuyang Yan [a], Yuyan Wu [a] and Dezhi Zeng [a]
[a] *School of Computer Science and Engineering, Chongqing University of Technology, Chongging, China*

Abstract. In existing methods for entity-relation extraction, both entity-driven and relation-driven approaches commonly suffer from insufficient interaction between entities and relationships. Specifically, there is a lack of utilization of the semantic information inherent in relationships. This paper proposed a relation semantic fusion-based entity relation extraction method (RSFnet). Firstly, all possible subjects are extracted from the sentence, and a mapping mechanism is used to obtain corresponding potential relations. At the same time, we treated relations as prior knowledge and used attention mechanisms to obtain sentence representations with relation semantics. The subject information is used as prior features, and the subject features are obtained through a bi-directional long-short term memory (BiLSTM) network. The updated sentence representations and enhanced subject features are further utilized for object and relation extraction, ultimately outputting triplets. The performance of the proposed model was validated through experimental results on three datasets. Additionally, this paper adopts convolutional encoding, resulting in better inference performance than methods based on Bidirectional Encoder Representations from Transformers (BERT), indicating that our model can improve triplet extraction performance while maintaining inference speed.

Keywords. relation semantic, mapping mechanism, prior knowledge, convolutional encoding

1. Introduction

The task of entity relation extraction aims to extract entity and relation facts from given text, forming relation triplets in the form of (subject(s), relation (r), object(o)). The extracted triplets can serve as the fundamental units of a knowledge graph, providing an external knowledge base for downstream tasks such as automatic summarization generation and dialogue generation.

Early approaches to entity relation triplet extraction often employed a pipeline method [1], which consisted of two stages: first, identifying all entities in the sentence, and then performing relation classification for each entity pair. Due to the error propagation issue, where errors in the first stage cannot be corrected in the second stage, this method had limitations. As a result, subsequent research proposed joint learning meth-

[1] Corresponding Author: Ling Lu, Chongqing University of Technology, E-mail:ll@cqut.edu.cn

ods for entities and relations, including feature-based models [2] and neural network-based models [3]. Among them, neural network-based models achieved excellent results in joint learning by using learned representations instead of manually constructed feature representations, gradually becoming the mainstream approach. For example, Wei et al. [4] proposed a cascaded joint extraction model that first extracts subjects and then jointly extracts relations and objects. The method incorporates subject information into the object and relation extraction stage, improving the performance of triplet extraction through the interaction of two tasks and effectively addressing the issue of entity overlap. Joint extraction methods that guide entity extraction with relations have also received significant attention. For example, Wang et al. [5] proposed a multi-hop attention-based entity-relation joint extraction method. It first labels the head entity and outputs multiple related tail entities. Then, it takes the tail entity as the next hop's head entity for input and iteratively performs relation extraction until the final entity relations are outputted. This method fully utilizes the latent relations between entities and enhances the performance of complex multi-hop relation extraction. Dai et al. [6] added a relation label embedding mechanism to the entity extraction layer, integrating text with relation labels. They utilized the subject's position information to selectively match suitable entity relation using attention, thereby improving precision. Although joint extraction methods are widely used due to their better interpretability and good experimental performance, existing methods still have shortcomings in utilizing and interacting with information within triplets, especially in terms of semantic representation of relations and utilization of subject information.

In response to the issues of insufficient information utilization and semantic information loss in existing joint extraction models, this paper proposes an entity-relation extraction method that integrates relation semantics. The main points of this paper's approach are twofold.

Firstly, we believe that by guiding entity extraction with subjects and relations, we can effectively control the redundancy of object extraction. This is because relations in the text generally depend on entities, and the number of relations is usually not much greater than the number of entities. For example, in the sentence "Arhus Airport is located in Tirstrup, Denmark.", the entities include "Arhus Airport", "Tirstrup" and "Denmark" but there is only one relation "located in". Given the known subject "Arhus Airport", we can extract the object "Tirstrup" through the relation "located in" while the unrelated entity "Denmark" will not be extracted.

Secondly, we believe that relations in triplets also contain rich contextual semantics. Integrating the semantic information of relations is beneficial for accurately extracting entities, which has not been widely addressed in existing research. For example, in the sentence "Peter is eating apples while watching TV.", we heuristically assume that since the conditional probability $P(eating \mid apples)$ is much larger than $P(eating \mid TV)$, the posterior probability $P(apples \mid eating)$ is greater than $P(TV \mid eating)$. In other words, perceiving the relation "eating" is helpful for extracting the object "apples". This viewpoint is even more useful in the case of implicit relations. For example, in Figure 1, the triplet in Sentence 2 is (Biden, president_of, United States),where the subject "Biden" has two candidate objects, "China" and "United States". We believe that without the semantic understanding of the implicit relation "president_of", it would be nearly impossible to correctly extract the object "United States".

Figure 1. Example of triplets extracted in implicit relation extraction.

In conclusion, this paper proposes a entity-relation extraction method called RSFnet (Relation Semantic Fusion-based Network) that integrates relationship semantics. Firstly, a convolutional encoder is used to identify the subject, and a mapping mechanism is used to obtain a possible set of relationships. Then, semantic fusion of relationships and entity information enhancement are carried out to further identify the corresponding objects and obtain the final triplet. The main work of this article is as follows:

1.Proposed a triplet extraction framework that integrates relation semantics. By calculating the attention of each word and the global representation under different relations, the relation information is integrated into the sentence, enriching the relations between entities and enhancing the interaction between relations and entities, thereby improving entity extraction performance.

2.Proposed a method for enhancing subject information representation. By integrating subject information with relative positional information and leveraging a BiLSTM network for deeper feature extraction, the subject representation is updated to further identify the corresponding object.

3.Experiments conducted on two datasets, NYT10 and NYT11, achieved performance superior to the baseline. Experiments involving the swapping of subject and object extraction order demonstrate the robustness of the proposed method. Additionally, the use of convolutional encoding in this paper allows RSFnet to outperform the baseline method in terms of inference performance and results.

2. Related Work

2.1. Joint extraction framework

Early methods for entity relation extraction often employed a pipeline approach, which separated entity recognition and relation extraction into two independent tasks. This method ignored the mutual relation between entities and relations, becoming a bottleneck in performance improvement. Subsequently, feature-engineering-based joint extraction methods addressed the interaction issue, but heavily relied on NLP tools for feature acquisition, requiring significant human effort and domain knowledge while also having error propagation issues. Later, with the excellent feature learning capability of neural networks, neural network-based methods gradually found applications in joint extraction.

Miwa et al. [7] proposed a parameter-sharing-based joint extraction method, decomposing joint extraction into different subtasks, where BILSTM and feed-forward neural networks were used for each subtask, reducing the complexity of feature learning. However, entities and relations were still extracted separately. A cascaded binary tagging framework was proposed in reference [6], which effectively resolved the issue of entity overlap while incorporating subject information into the object and relation extraction stages, improving triplet extraction performance. However, this method simply added the subject to the original text information, resulting in a single form of interaction between the two sub-tasks and the problem of relation redundancy. To address the issue of relation redundancy, Li et al. [8] introduced a mapping mechanism from entity types to predefined relations. This mechanism avoided the need to iterate through all relations when predicting overlapping relations, reducing a significant amount of meaningless computations. Zheng et al.[9] proposed a joint triplet extraction framework based on latent relations and global correspondence, which greatly alleviated the problems of redundant relation judgments, poor generalization of span-based extraction, and low efficiency of subject-object alignment. However, this framework still had limitations in terms of utilizing relation information and interaction between sub-tasks. Zhe et al. [10] proposed an end-to-end relation-first blank filling network. The model encoded prior knowledge of relations in templates and transformed relations into specific relation templates using the semantic information of relations. Finally, it extracted entity pairs through blank filling, effectively improving triplet performance. This approach provides inspiration for subsequent research.

Overall, neural network-based joint extraction methods have gained attention for effectively addressing issues such as error propagation, subtask interaction, and information redundancy. The approach presented in reference [10] used relations as prior knowledge to guide entity extraction, providing insights into subtask interaction. This inspired the method proposed in this paper to enhance text representation by leveraging relational semantic information and establish interaction between entities and relations, further improving triplet extraction performance.

2.2. Method of interaction between triplet relations and entity information

The interaction between entity pairs and between entities and relations is crucial in joint extraction methods. Zhang et al. [11] applied a local focusing mechanism to entity pairs and their corresponding contexts to obtain richer feature representations from local contexts, thereby completing the Relation Extraction (RE) task. Zheng et al. [12] used a weighted relative position attention mechanism to modify the vanilla Transformer encoder, which flexibly captured the semantic features between entities. Sun et al. [13] proposed a recurrent interaction network that allows explicit dynamic interaction between entity recognition and relation extraction tasks, capturing the mutual correlation between them. Yuan et al. [14] utilized an attention gate mechanism to obtain fine-grained semantic representations for specific relations, greatly improving the impact of relevant relation types on entity recognition. These studies demonstrate that better interaction between entity and relation information can be achieved through various approaches.

Inspired by these findings, the present study enhances the interaction between entity and relation information by utilizing attention mechanisms. By integrating subject information with relative positional information and applying further deep-level feature extraction through a BiLSTM network, the representation of the subject is updated. This updated representation is then fused with the sentence representation containing relation information and the subject feature representation. This fusion is used to guide entity extraction, thereby improving the performance of triplet extraction.

2.3. Convolutional Encoder

Currently, many models adopt Transformer-based pre-trained language model encoders, which has a powerful ability to capture long-distance dependencies and contextual semantic features. However, it also increases memory consumption, limiting model training and inference time. Convolutional neural networks have been found to effectively extract text features and explore associations between words. Yu et al. [15] proposed a method using dilated gated convolutional neural networks, which increased the mutual correlation between distant words. Compared to the BERT model, this model achieved optimal results while being lightweight and fast. Therefore, in reference [8], a convolutional encoding structure combining dilated convolution, gating units, and residual connections was designed to improve the computational efficiency of the encoder. Hence, this paper utilizes this convolutional structure as the encoder, not only reducing training and inference time but also ensuring performance requirements.

3. Methodology

The overall framework of RSFnet, as proposed in this paper, is depicted in Figure 2. It is a cascaded binary tagging framework that integrates relationship information. The framework consists of four components: an encoding layer, entity extractor A, information fusion layer, and entity extractor B.Entity extractors A and B are responsible for extracting two entities. For example, entity extractor A can be used to extract the subject, while entity extractor B can be used to extract the object. The reverse is also possible.The information fusion layer comprises two parts: the fusion of relationship semantics and the enhancement of entity features.The encoding layer encodes the input sentence and obtains sentence-level feature vectors using self-attention. Then, entity extractor A is employed to extract one entity, referred to as entity A, from the sentence.The relationship semantic fusion layer incorporates a mapping mechanism [8] to extract the relevant relationship set for entity A and calculates the semantic representation of all relationships. This representation is then fused into the sentence representation to update it. Finally, the updated sentence representation is combined with the enhanced entity features. Entity extractor B is utilized to further extract entity B, resulting in the output of a triple.

3.1. The Encoder Layer

The model takes a sentence $X = [x_1, x_2, ..., x_n]$ as input, where $x_i \in \mathbb{R}^d$ represents the word embedding of the i-th word. The embedding dimension is denoted as d, and n represents the number of words in the sentence. The word embeddings x_i are composed of GloVe word embeddings X_g and trainable position embeddings X_p. The encoder adopts a convolutional encoder proposed in reference [8]. The encoder consists of L stacked blocks $Block(\cdot)$, and each block contains two dilated convolutions with a dilation rate of d_i, a gating unit, and a residual connection. The dilated convolution is represented as $DilatedConv(\cdot)$. By passing the sentence through the encoding layer, the representation of the sentence $H = [w_1, w_2, ..., w_n]$ is obtained as follows:

$$\mathbf{H} = \text{Block}(\cdots(\text{Block}(\mathbf{X}))) \tag{1}$$

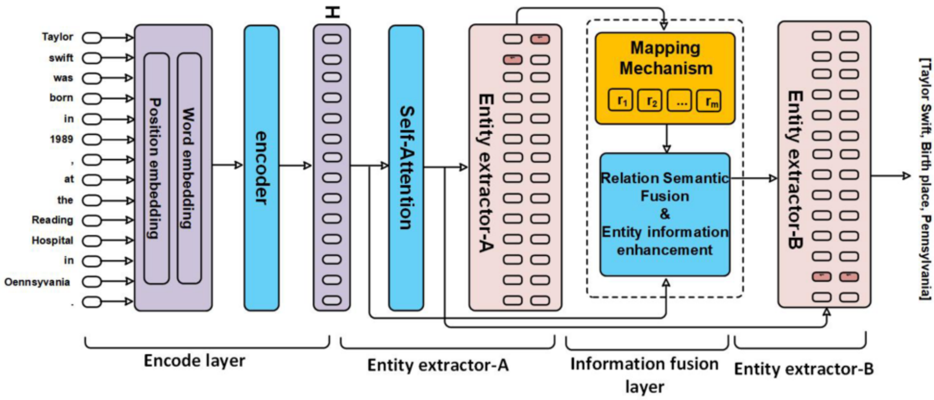

Figure 2. The overall structure of RSFnet.

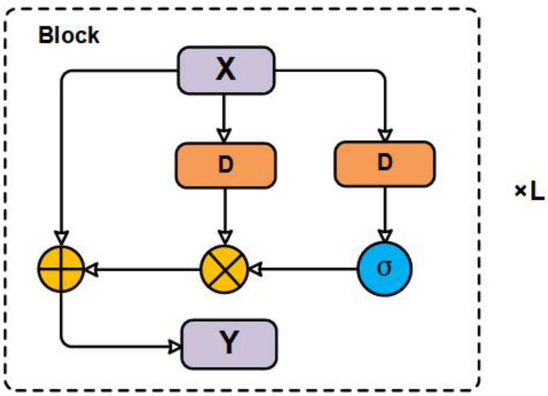

Figure 3. Structure of the Convolutional Encoder.

$$Y_a = DilatedConv_a(X) \tag{2}$$

$$Y_b = DilatedConv_b(X) \tag{3}$$

$$Y_i = Y_a \otimes sigmoid(Y_b) + X \tag{4}$$

Where $H \in \mathbb{R}^{n\times d}$, $w_i \in \mathbb{R}^d$ represent the encoded contextual representation of the i-th word, $\otimes$ denoted as element-wise multiplication. The output of the i-th stacked block and the input of the (i+1)-th stacked block are represented by Y_i, and the final representation of the sentence is denoted as Y_L, which is the output of the last stacked block. The overall structure of the convolutional encoder is depicted in Figure 3.

Since the convolutional encoder shares a significant number of parameters and treats each word x_i in the sentence equally, we introduce multi-head self-attention [16] to generate auxiliary entity features. The entity A feature representation H_h is given as follows:

$$\mathbf{H}_h = \mathrm{softmax}(\frac{\mathbf{QK}^{\mathrm{T}}}{\sqrt{d_k}})\mathbf{V} \tag{5}$$

$$\mathbf{Q} = \mathbf{W}_q \cdot \mathbf{H} + b_q \tag{6}$$

$$\mathbf{K} = \mathbf{W}_k \cdot \mathbf{H} + \mathrm{b}_k \tag{7}$$

$$\mathbf{V} = \mathbf{W}_v \cdot \mathbf{H} + b_v \tag{8}$$

Where $d_k = d$ represents the dimension of the attention key vector. The weights and biases for the query, key, and value, denoted as $W_q, W_k, W_v \in \mathbb{R}^{d\times d}$ respectively, are used to obtain the values of $b_q, b_k, b_v \in \mathbb{R}^d$.

3.2. Entity Extractor A

To extract entity A using the entity extractor A, we concatenate the sentence representation H with the entity A feature, resulting in $[[w_1, w_1^h], ..., [w_n, w_n^h]]$. We have a set of relation types $T = \{t_1, t_2, ..., t_m\}$. We calculate the score of the i-th word being the start and end positions of an entity of type t_j as follows:

$$o_{ij}^{hs} = \mathbf{W}_{ij}^{hs} \cdot [\mathbf{w}_i, \mathbf{w}_i^h] + b_{ij}^{hs} \tag{9}$$

$$o_{ij}^{he} = \mathbf{W}_{ij}^{he} \cdot [\mathbf{w}_i, \mathbf{w}_i^h] + b_{ij}^{he} \tag{10}$$

Where o_{ij}^{hs} represents the score of the i-th word belonging to the entity A with respect to the starting position for type t_j. .

The threshold for entity boundaries can be different from the thresholds used for other words. An adaptive threshold strategy [17] is employed to improve the accuracy of entity labeling and enhance the model's generalization capability. Only when the score of an extractor exceeds its position-related threshold, the corresponding position is marked as 1. For all starting positions p_{ij}^s of entity A at position $i \in [1, n]$, we denote positive instances (positions expected to be marked as 1) as P_i and negative instances as N_i. An AT class is introduced to store all starting positions related to the AT type, where $P_{i;AT}$ represents the threshold at position i. During training, the loss for starting position labeling is defined as follows:

$$L_h^s = -\sum_{i=1}^{n} \sum_{p_{ij}^s \in N_i} \log(\frac{\exp(o_{ij})}{\sum_{p_{ik}^s \in P_i \cup \{p_{i;TH}\}} \exp(o_{ik})}) - \sum_{i=1}^{n} \log(\frac{\exp(o_{i;TH})}{\sum_{p_{ik}^s \in N_i \cup \{p_{i;TH}\}} \exp(o_{ik})}) \tag{11}$$

Where $o_{i;AT}$ represents the threshold for position i. Similarly, L_h^e denotes the threshold for marking the end position of the entity, and it follows the same loss function as L_h^s. The label loss for entity A, denoted as L_h, is the sum of the loss for the start position and the loss for the end position:

$$L_h = L_h^s + L_h^e \tag{12}$$

3.3. Information Fusion Layer

The information fusion layer is responsible for re-encoding the sentence representation and the subject information, aiming to further extract relationships and objects. It consists of two parts: relationship semantic fusion and entity feature enhancement.

Relation Semantic Fusion After identifying all possible entities A, in order to capture all entities B under the relationship $R = \{r_1, r_2, ..., r_j\}$, the relationship information is integrated into the sentence to enrich its information representation. The relationship information is obtained by leveraging GloVe embeddings [18] for the relationship R. Since each word in the sentence plays a different role in relation to different entities B, an attention mechanism is used to measure the attention scores for each relationship r_j with respect to the global representation h_g of the sentence and the representation of each word x_i. By taking the weighted sum of the sentence words based on the attention scores, a sentence representation under the relationship type j is generated as follows:

$$h_g = avg\{x_1, x_2, ..., x_n\} \tag{13}$$

$$e_{ij} = \mathbf{v}^{\mathbf{T}} \tanh(\mathbf{W}_{\mathrm{r}}\mathbf{r}_j + \mathbf{W}_g\mathbf{h}_g + \mathbf{W}_x\mathbf{x}_i) \tag{14}$$

$$a_{ij} = \mathrm{softmax}(e_{ij}) \tag{15}$$

$$c_j = \sum_{i=1}^{l} a_{ij}\mathrm{x}_i \tag{16}$$

Where n is the length of the sentence, h_g is the global representation of the sentence, e_{ij} represents the attention scores obtained using a multi-layer perceptron (MLP). It denotes the importance of each word and the global sentence representation for different relations. The MLP includes a hidden layer with a tanh activation function. r_j represents the relation embedding. a_{ij} is the weight coefficient calculated through softmax, and the specific sentence representation c_j is obtained through weighted averaging.

Entity Feature Enhancement In addition to incorporating relation embeddings,we further determine a potential set of relations R' based on the current entity type using a type-relation mapping mechanism. Finally, the sentence representation is updated by fusing the weighted connection of the sentence representation c_j with the original sentence representation w_i. The final representation of the i-th word is obtained by setting the entity embedding layer $V_t \in \mathbb{R}^{K \times d_t}$ and the relative position embedding layer $V_p \in \mathbb{R}^{n \times d}$. The start and end features W_a, W_b of entity A are obtained from H,the type features w^t of entity A are derived from V_t, and the relative position features w_a^p and w_b^p are derived from V_p, and then $(w_a + w_a^p), (w_b + w_b^p)$ and w^t are connected. BILSTM model is used to further extract features to form entity A features w^h:

$$w_i' = c_j + w_i \tag{17}$$

$$w^h = BILSTM(w^t; W_a + w_a^p; W_b + w_b^p) \tag{18}$$

Where w_i' represents the new sentence representation after weighted connection, and w^h represents the enhanced feature representation of the entity.

3.4. Entity Extractor B

Finally, to further extract entity B, the merging of entity A's feature information and the updated sentence information is performed to assist in more accurate identification of entity B.We concatenate the sentence representation H, the auxiliary feature H_t for entity B, and the entity A feature w^h to form $[[w_1', w_1^t, w^h, ..., [w_n', w_n^t, w^h]]$. Therefore, we compute scores for the i-th word as the starting and ending positions of entity B with potential relation $r_j \in R' \subset R$.

$$o_{ij}^{t_s} = W_{ij}^{t_s} \cdot [w_i', w_i^t, w^h] + b_{ij}^{t_s} \tag{19}$$

$$o_{ij}^{t_e} = W_{ij}^{t_e} \cdot [w_i', w_i^t, w^h] + b_{ij}^{t_e} \tag{20}$$

Where $o_{ij}^{t_s}$ represents the score of the i-th word as the starting position of entity B with relation r_j. Similar to the loss for entity A labeling, we directly calculate the loss value L_t for entity B labeling using Eq. (12). The final loss value is given by Eq.(21)

$$L = \frac{1}{|D|}(\sum_{S_i \in D} \sum_{h_j \in Z_i} L_{h_j} + \sum_{S_i \in D} \sum_{t_j \in Z_i} L_{t_j|h}) \tag{21}$$

Where D represents all the sentences, and Z_i represents all the relation triplets in sentence S_i.

4. Experiments

4.1. Datasets

The experiments in this paper were conducted on three datasets: NYT10[19], NYT11[19], and NYT24[20].The entity types [PER], [LOC], [ORG] and [OTH] were used to indicate four entity types and the relation mapping mechanism. We utilize the mapping mechanism to identify potential relations. NYT10 and NYT11 are versions generated by aligning the original data from the New York Times corpus with Freebase (a knowledge graph database). NYT10 is a smaller version, and its test set is manually annotated. NYT24 is created by filtering out sentences with more than 100 words and sentences containing non-positive class triples from the NYT dataset. Then, 5000 sentences were randomly selected as the test set, 5000 sentences as the validation set, and the remaining 56195 sentences as the training set. The composition is shown in Table 1.

Table 1. Statistics of NYT10, NYT11 and NYT24 datasets.

Dataset	#Relation	Train	Valid	Test
NYT10	29	70339	-	4006
NYT11	12	62648	-	369
NYT24	24	56196	5000	5000

Table 2. Experimental results of different methods on the NYT10, NYT11 and NYT24 datasets. † denotes results generated using the source code provided in the original paper, while other results are retrieved from the original paper.

Models	**NYT10**			**NYT11**			**NYT24**		
	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1
CasRel†[4]	78.0	69.0	73.2	50.3	58.1	53.9	89.9	89.1	89.5
TPLinker†[21]	80.1	66.4	72.6	**56.2**	55.1	55.7	91.0	91.8	91.4
PRGC†[9]	80.2	66.5	72.7	54.4	56.3	55.3	89.9	90.9	90.4
CopyRE[3]	45.2	56.9	50.4	34.7	53.4	42.1	61.0	56.6	58.7
WDec[22]	**84.6**	62.1	71.6	-	-	-	**94.5**	76.2	84.4
SPN†[23]	79.5	67.1	72.8	52.7	55.4	54.0	93.3	91.8	92.5
HRL[19]	71.4	58.6	64.4	53.8	53.8	53.8	-	-	-
GRTE†[24]	79.8	67.6	73.2	53.6	58.2	55.8	92.5	**92.7**	**92.6**
FastRE[8]	78.0	**70.1**	73.8	54.1	**58.7**	56.3	89.6	86.3	87.9
RSFnet(ours)	80.3	68.7	**74.1**	56.0	58.1	**57.0**	88.2	87.5	87.9

4.2. Settings

For the parameters used in the experiments of this paper, the convolutional encoder consists of 6 stacked blocks with convolution rates of 1, 2, 4, 1, 1 and 1. The training process utilizes the Adam optimizer with a learning rate of 1e-3 and a batch size of 32. All hyperparameters were adjusted on the validation set. The experiments were conducted on an NVIDIA GeForce RTX 3060 GPU. The effectiveness of the model is validated by calculating the precision (P), recall (R), and F1 score for extracting entity-relation triplets during the experiments.

In experiments 5.1, 5.2, and 5.3, we used extraction mechanism A to extract the subject and extraction mechanism B to extract the object. However, in experiment 5.4, we used extraction mechanism A and B to extract the object and subject, respectively, to validate the robustness of RSFnet.

5. Experimental Results and Analysis

5.1. Comparison with Existing Methods

We compared our model with state-of-the-art relation extraction models: (1) Using labeling methods to accomplish the triplet extraction task, such as CasRel[4], TPLinker[21], PRGC[9], FastRE[8]. (2) Transforming the relationship extraction task into a generation task based on generative models, such as CopyRE[3], WDec[22], SPN[23]. (3) Extracting triplets using reinforcement learning methods, such as HRL[19], GRTE[24].

Table 3. Statistics of NYT10, NYT11 and NYT24 datasets.

Models	P	R	F1
RSFnet	56.0	**58.1**	**57.0**
w/o Relational semantic fusion	55.4	58.1	56.7
w/o Entity feature enhancement	**56.2**	56.5	56.3

The comparative experimental results are presented in Table 2. It can be observed that RSFnet, proposed in this paper, achieves superior results compared to the baselines on the NTY10 and NYT11 datasets. For the NYT10 dataset, our method achieves an F1 score of 74.1%, which is a 0.3% improvement in F1 score over FastRE. On the NYT11 dataset, the F1 score for entity-relation triplet extraction is 57.0%, which is a 0.7% improvement compared to FastRE.On the NYT24 dataset, the proposed method achieves a comparable performance level to FastRE. The experiments demonstrate that the method presented in this paper has a positive impact on improving the overall performance of relation extraction.

5.2. *Ablation Study*

To evaluate the impact of the relation embedding and entity feature enhancement modules on performance, this paper conducted ablation experiments on the NYT11 dataset, and the results are shown in Table 3. "w/o Relational semantic fusion" indicates the removal of the relational semantic fusion module, while "w/o Entity feature enhancement" indicates the removal of the entity feature enhancement module.

From Table 3, it can be observed that both ablation experiments result in a performance decline on the NYT11 dataset. We believe that relation information contains rich semantic information, and when fused with the sentence, it can capture fine-grained semantic representations that are beneficial for subsequent object extraction. The absence of relational semantic fusion leads to a decrease in overall performance. When the entity feature module is removed, there is a decrease of 0.7 percentage points, indicating that enhancing the subject representation through the fusion of entity features and relative positional information using the BiLSTM network can effectively utilize the subject features to assist in object extraction, resulting in more accurate triplet extraction.

5.3. *Performance Analysis*

To evaluate the inference efficiency of RSFnet, a comparison was made with a model using the BERT encoder, and the results are shown in Table 4. In the table, "Param" represents the number of model parameters obtained through the official implementation with default configurations. "Train" and "Infer" respectively indicate the total training time (in minutes) and the total inference time (in seconds).

As can be seen, RSFnet utilizes fewer parameters in the GloVe word embeddings. Its use of a convolutional encoder significantly reduces the computational pathways, resulting in shorter training and inference times compared to other models.

5.4. *Study on Robustness*

To further validate the stability of the RSFnet method, entity extractors A and B were used to extract subjects and objects (s→o), as well as objects and subjects (o→s). A

Table 4. Speed comparison experiments with other encoder models on the NYT10, NYT11 and NYT24 datasets.

Models	NYT10			NYT11			NYT24		
	Param	Train	Infer	Param	Train	Infer	Param	Train	Infer
CasRel	107,729K	251m	266s	107,698K	448m	24s	107,720K	420m	327s
TPLinker	109,606K	984m	191s	109,548K	973m	16s	109,603K	885m	235s
SPN	141,754K	516m	202s	104,648K	380m	19s	141,429K	473m	254s
PRGC	108,931K	290m	134s	108,891K	267m	12s	108,919K	272m	161s
GRTE	119,450K	890m	176s	119,450K	843m	17s	119,387K	795m	221s
Ours	8361K	203m	66s	8356K	178m	5.5s	8359K	164m	83s

Table 5. Robustness experiments to verify subject-first extraction and subject-first extraction on NYT10, NYT11 and NYT24 datasets.

Models	NYT10			NYT11			NYT24		
	P	R	F1	P	R	F1	P	R	F1
FastRE	78.0	70.1	73.8	54.1	58.7	56.3	89.6	86.3	87.9
(s→o)	80.3	68.7	**74.1**	56.0	58.1	**57.0**	88.2	**87.5**	87.9
(o→s)	**80.4**	68.6	74.0	**56.9**	57.0	56.6	88.6	87.1	87.9

comparative experiment was conducted against FastRE, and the results are shown in Table 5. It can be observed that RSFnet consistently maintains stable performance in both the (s→o) and (o→s) processes.

6. Conclusion

This paper addresses the limitations of existing methods for joint extraction of entity and relationship triplets, including insufficient interaction between entity extraction and relationship extraction stages and underutilization of semantic information about relationships. To tackle these issues, a novel joint extraction method called RSFnet is proposed. RSFnet first extracts all possible subjects and then employs a mapping mechanism to obtain a set of potential relationships. It utilizes an attention mechanism to capture the semantic information of relationships and combines it with enhanced subject features for the joint extraction of objects and relationships. This process yields the final triplets. Experimental results demonstrate that RSFnet achieves higher F1 scores compared to baseline methods on the NYT10 and NYT11 datasets. Furthermore, RSFnet exhibits superior inference time efficiency compared to BERT models.

References

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

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

[3] 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 (V olume 1: Long Papers), volume 1, pages 506–514.

[4] Zhe pei Wei, Jianlin Su, Yue Wang, and et al. A novel cascade binary tagging framework for relational triple extraction. In ACL, 2020.
[5] Wang H, Wu Y T. Joint extraction of entity relationships based on multi-hop attention and its application [J]. Proceedings of the Journal of Taiyuan University of Technology, 2022, 53(1):63-70.
[6] Dai Jianfeng, Chen Xingyu, Dong Ligang, et al. Joint Extraction Method of Triplets Combining Hybrid Embedding and Relation Label Embedding [J]. Telecommunications Science, 2023, 39(2): 132-144.
[7] Miwa M, Bansal M. End-to-end relation extraction using LSTMs on sequences and tree structures[C]//Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Aug 7-12, 2016. Stroudsburg: ACL, 2016: 1105-1116.
[8] Li, G., Chen, X., Wang, P., Xie, J., & Luo, Q. 2022. FastRE: Towards Fast Relation Extraction with Convolutional Encoder and Improved Cascade Binary Tagging Framework. ArXiv, abs/2205.02490.
[9] Hengyi Zheng, Rui Wen, Xi Chen, and et al. PRGC: Potential relation and global correspondence based joint relational triple extraction. In ACL, 2021.
[10] Zhe Li, Luoyi Fu, Xinbing Wang, Haisong Zhang, and Chenghu Zhou. 2022. RFBFN: A Relation-First Blank Filling Network for Joint Relational Triple Extraction. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, pages 10–20, Dublin, Ireland. Association for Computational Linguistics.
[11] Zhang, H., Zhang, G. & Ma, R. Syntax-informed self-attention network for span-based joint entity and relation extraction. Appl.Sci. 11(4), 1480 (2021).
[12] Zheng, W., Wang, Z., Y ao, Q.&Li, X.Wrtre: Weighted relative position transformer for joint entity and relation extraction. Neurocomputing 459, 315–326 (2021).
[13] Sun K, Zhang R, Mensan S, et al. Recurrent interaction network for jointly extracting entities and classifying relations[C]//Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, Nov 16-20, 2020. Stroudsburg: ACL, 2020: 3722-3732.
[14] Y . Yuan, X. Zhou, S. Pan, Q. Zhu, Z. Song, and L. Guo, "A relation-specific attention network for joint entity and relation extraction," in International Joint Conference on Artificial Intelligence 2020, pp. 4054–4060, Association for the Advancement of Artificial Intelligence (AAAI), 2020.
[15] Yu, Fisher , and V. Koltun . "Multi-Scale Context Aggregation by Dilated Convolutions." ICLR 2016.
[16] Ashish V aswani, Noam Shazeer, Niki Parmar, and et al. Attention is all you need. In NIPS, 2017.
[17] Zhou, W., Huang, K., Ma, T., & Huang, J. (2020). Document-Level Relation Extraction with Adaptive Thresholding and Localized Context Pooling. ArXiv, abs/2010.11304.
[18] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In EMNLP, 2014.
[19] Ryuichi Takanobu, Tianyang Zhang, Jiexi Liu, and Minlie Huang. A hierarchical framework for relation extraction with reinforcement learning. In AAAI,2019.
[20] Xiangrong Zeng, Daojian Zeng, Shizhu He, and et al. Extracting relational facts by an end-to-end neural model with copy mechanism. In ACL, 2018.
[21] 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.
[22] Tapas Nayak and Hwee Tou Ng. Effective modeling of encoder-decoder architecture for joint entity and relation extraction. In AAAI, 2020.
[23] Dianbo Sui, Y ubo Chen, Kang Liu, and et al. Joint entity and relation extraction with set prediction networks. In CoRR, 2021.
[24] Feiliang Ren, Longhui Zhang, Shujuan Yin, and et al. A novel global feature-oriented relational triple extraction model based on table filling. In EMNLP,2021.