Joint Extraction of Interaction Features and Attention Mechanism Fused with Entity-Relations

Lisha Su, Rong Yan*
College of Computer Science, Inner Mongolia University
Inner Mongolia Key Laboratory of Multilingual Artificial Intelligence Technology
National & Local Joint Engineering Research Center of Intelligent Information Processing Technology for Mongolian
Hohhot 010021, China
Email: csyanr@imu.edu.cn

***Abstract*—Joint extraction of entities and relations is an essential task in information extraction. Recently, tagging-based models have gained attention but with poor performance on overlapping triplets, which confronted the issues like cascading errors, information redundancy, and sensitivity to subject extraction. To solve the issue of overlapping triplets, we propose a model combined Gated Attention Unit (GAU) and Multi-head Attention Mechanism (MHA) in this paper. This model distinguishes features for entities and relations, and performs entity recognition for each head entity, different head entities are matched to the same or different tail entities. Then specific tail entity tagging is applied to determine all possible relations and corresponding tails. At the same time, in order to better solve the long-distance dependence and complex multiple relations between entities, GAU and MHA are used to enhance the semantic representation, effectively identify entities and relations, and enhance the learning ability and generalization ability of the model. Experiment results on the public datasets DuIE2.0 and CMeIE show that our model performs better than baselines.**

***Index Terms*—joint extraction, entity tags, overlapping triplets, entity nesting**

I. Introduction

As one of the crucial tasks in information extraction, joint extraction of entities and relations plays a pivotal role in numerous downstream applications of natural language processing, encompassing knowledge question answering, information retrieval, and recommendation systems. It is designed to identify all entities in unstructured text, automatically infer potential semantic relations between these entities to extract relational triples (head, relation, tail).

In the joint extraction of entities and relations, most existing joint models cannot effectively deal with the problem of overlapping triples, Zeng et al. [1] were the first to tackle the issue of overlapping triplets. Sentences with different types of overlap can be classified into three categories: Normal, Single Entity Overlap (SEO), and Entity Pair Overlap (EPO). Addressing overlapping triplets is an important problem for joint extraction, which is essential for applications, including information retrieval, automatic summarization, and machine translation.

Currently, the joint extraction method is the mainstream method [2]–[4] for relational triple extraction. It utilizes the one model to perform both entity recognition and relation extraction tasks, thereby enhancing extraction efficiency. Moreover, the joint extraction method based on tagging-based models [5], [6] is a good way to solve overlapping triples. Wei et al. [5] introduced a new binary tag structure to identify the subject in the sentence, and then extracted the relation and object based on the identified subject. However, this method relies too much on the extraction of the subject and neglects the semantic relation between the subject and other words. Chen et al. [7] first detected potential relations in sentences to avoid relational redundancy, and then performed entity recognition for each specific relation to solve the overlapping triplet problem. Tagging-based models can effectively deal with overlapping triples by designing specific tags that directly link relations and entities. However, the extraction effect of this method depends on the quality of the specific labels on the data.

In order to deal with overlapping triples effectively, we propose a method based on Gated Attention Unit (GAU) and Multi-head Attention Mechanism (MHA), through the recognition of each extracted head entity, different head entities are matched to the same or different tail entities, and specific tail entity tagging is applied to determine all possible relations and corresponding tail entities. GAU and MHA enhance semantic representation, improve the model's ability to identify entities, and relations. In this paper, our model uses the pointer annotation method to extract the relational triples, and extracts the relationship triples according to the labeling results to improve the extraction performance of the model.

II. Relate work

The current popular joint extraction of entities and relations models can be divided into three categories, which are sequence-to-sequence models, table-filling models and tagging-based models.

A. Sequence-to-sequence models

Sequence-to-sequence models convert a relational triple extraction into a generation task by generating triples in a specific order, first forming relations and then generating entities. Then, the researchers used the sequence-to-sequence model to solve the overlapping triplet issue in the extraction

DOI reference number: 10.18293/SEKE2024-119

task [8]. The REBEL model [9] converted the extraction task into a language generation task by representing the triplet as a sequence of text. However, sequence-to-sequence models are not conducive to long-distance relation extraction, especially when dealing with sentences with a large number of overlapping triples, and the efficiency needs to be improved.

B. Table-filling models

Table-filling models use a two-dimensional table to joint extraction of entities and relations, which can avoid error propagation and effectively handle many-to-many relations and overlapping triples. Yan et al. [10] introduced a partitioned filtering network to learn the respective feature representations of entity recognition tasks, and relation classification tasks, then identifying relations triples in a table-filled manner. Ning et al. [4] proposed a vertex-based bounding box detection, coupled with auxiliary global relation triplet region detection, to ensure that the region information of the triplet was fully utilized. Nevertheless, table-filling models face efficiency and storage challenges when dealing with large-scale data.

C. Tagging-based models

Tagging-based models support joint extraction of entities and relations by adding specific tags or tags to the input sentence, which can be used to indicate the location of entities in the sentence and the relations between them. Ren et al. [11] proposed a two-way extraction framework, which extracted entities and relations from two perspectives: subject-based extraction and object-based extraction, respectively, and finally added a double affine model to classify the relation between entity pairs to improve the accuracy of model extraction. Whereas, the extraction effect of tagging-based models depend on the quality of a particular label of the data.

III. Methodology

The overall structure of the joint extraction of entities and relations model proposed in this paper is shown in fig. 1, which consists of an encoder and a two-step decoder. The encoder is used to obtain the feature representation of the text. The decoder contains two back-and-forth cascade steps, the head entity tagging module and the tail entity tagging module. Our model applies different features to entities and relations, to detect the relations and entities contained in the input sentences, then tags entities based on head entity tagging module, and the tail entity tagging module, and adds MHA and GAU components to the two modules to enhance the semantics and improve the model extraction ability.

A. Task definition

Given the text X=$\{x_1, x_2, \cdots, x_l\}$ contains l tokens, the goal of relational triple extraction task is to extract sentences for all potential triples Y as shown in (1):

$$Y = \{(h, r, t) | h, t \in \varepsilon, r \in R\} \tag{1}$$

where ε represents the pair of entities drawn into the triplet; R=$\{r_1, r_2, \cdots, r_j\}$ represents the set of relations containing j relations, r represents the relation between the head and tail entities, respectively, and h and t represent the head and tail entities.

B. BERT encoder

BERT [12] is a popular pre-training language model based on the multi-layer bidirectional transformer, which can comprehensively represent the semantic information of the context. Most of existing state-of-the-art methods like CasRel [5] and TPLinker [13], they use an unified feature for subjects, objects, and relations. But we think that different kinds of items in triples have their own characteristics. And should be represented by different features. So we use BERT to generate two different representation sequences as head entities features h_s^i and relational category features h_r^i, respectively, and is calculated as shown in (2):

$$\begin{aligned} h_s^i &= W_s h^i + b_s \\ h_r^i &= W_r h^i + b_r \end{aligned} \tag{2}$$

where W_s, $W_r \in R^{d_h \times d_h}$ represent the trainable matrices, b_r and $b_s \in R^{d_h}$ represent the bias terms and d_h represents the dimensionality.

C. Head entity tagging module

Our head entity tagging method follows the binary tagging framework in CasRel [5]. Indicate whether the current tag corresponds to the start or end position of the topic by assigning a binary label (0 or 1). We have further refined the binary tagging framework in the entity tagging module, learning additional features through MHA and GAU, MHA and GAU components are required because MHA can learn enhanced text representation. At the same time, GAU can obtain richer and deeper semantic information, as shown in (3):

$$\begin{aligned} hs1_i &= GAU(h_s^i) \\ hs2_i &= MHA(h_s^i) \end{aligned} \tag{3}$$

for the end position of the head entity, the proposed model get two deeper vectors $ho1_i$ and $ho2_i$ in the same way as the start position. These deeper features ($hs1_i$, $hs2_i$, $ho1_i$ and $ho2_i$) are combined simultaneously for head entity tagging, as shown in (4):

$$\begin{aligned} p_s^{i,start} &= \sigma(W_s^{start}(W_1 hs1_i + W_2 hs2_i) + b_s^{start}) \\ p_s^{i,end} &= \sigma(W_s^{end}(W_3 ho1_i + W_4 ho2_i) + b_s^{end}) \end{aligned} \tag{4}$$

where $p_s^{i,start}$ and $p_s^{i,end}$ represent the probability that the $i\text{-}th$ tag in the input sentence sequence is recognized as the start and end position of the head entity, respectively. For each head entity, the same decoding process is applied repeatedly. W represents the weight matrix and b represents the bias term. σ denotes the sigmoid activation function in this article.

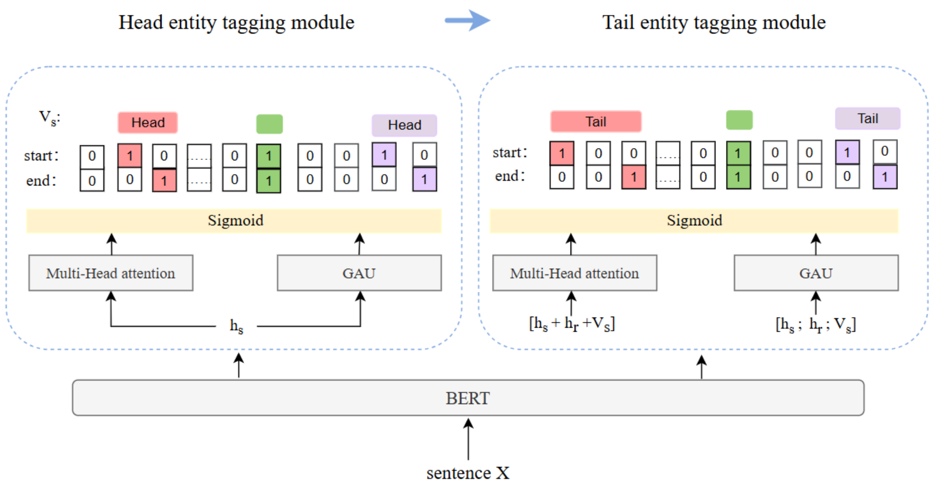


Fig. 1: The architecture of our model.

GAU represents a gated attention unit, which combines a gated linear unit and an attention mechanism, as shown in (5):

$$\begin{aligned} GAU &= (U \odot AV)W_o \\ U &= \phi_u(h_s^i W_u) \\ V &= \phi_v(h_s^i W_v) \\ Z &= \phi_z(h_s^i W_z) \\ A &= relu^2(Q(Z)K(Z)^T + b) \end{aligned} \tag{5}$$

where U, V and Z represent the shared representation, A represents the attention weight, Q and K represent the simple affine transformations (similar to LayerNorm), b represents the bias term, $\odot$ represents an element-by-element multiplication, ϕ represents an element-by-element activation function, and $relu$ represents an activation function.

MHA represents Multi-head attention Mechanism. After the MHA obtains the final representation of the encoder, the query matrix, key matrix, and value matrix are established, which are mapped to m different subspaces, and then input into m different parallel headers, as shown in (6):

$$\begin{aligned} MHA(Q,K,V) &= Concat(head_1, \ldots, head_m)W^o \\ head_i &= Attention(h_s^i W_i^Q, h_s^i W_i^K, h_s^i W_i^V) \end{aligned} \tag{6}$$

D. Tail entity tagging module

The internal structure of the tail entity tagging module is the same as that of the head entity tagging module, except in its input. In the tail entity tagging module, the $s\_k\text{-}th$ head entity vector v_s, the maximum pooling operation is carried out to extract the most significant head features as the subject information, as shown in (7):

$$v_s^{s\_k} = \max pool(h_s^{s\_k\_start}, \ldots, h_s^{s\_k\_end}) \tag{7}$$

We combine two feature fusion methods(adding and concatenating) to fuse the $i\text{-}th$ entity vector represents h_s, the $i\text{-}th$ relation vector represents h_r, and the $s\_k\text{-}th$ head entity vector represents v_s. Then the deeper semantic feature representations are obtained through different neural network modules MHA and GAU. These two deeper vectors $hs3_i$ and $hs4_i$, as follows the (8):

$$\begin{aligned} hs3_i &= MHA([h_s^i + h_r^i + v_s^{s\_k}]) \\ hs4_i &= GAU([h_s^i; h_r^i; v_s^{s\_k}]) \end{aligned} \tag{8}$$

These deeper features ($hs3_i$, $hs4_i$, $ho3_i$ and $ho4_i$) are combined simultaneously for tail entity tagging, as shown in (9):

$$\begin{aligned} p_o^{i,start} &= \sigma(W_o^{start}(W_1 hs3_i + W_2 hs4_i) + b_o^{start}) \\ p_o^{i,end} &= \sigma(W_o^{end}(W_3 ho3_i + W_4 ho4_i) + b_o^{end}) \end{aligned} \tag{9}$$

where $p_o^{i,start}$ and $p_o^{i,end}$ represent the probability that the $i\text{-}th$ token in the input sentence sequence is recognized as the beginning and end position of the tail entity, respectively.

IV. Joint Training

The loss function for the head entity tagging module is shown in (10):

$$\begin{aligned} L_{\text{head}} = -\frac{1}{l}\sum_{i=1}^{l} & (y_i^{\text{i,start}} \log p_s^{\text{i,start}} + (1 - y_i^{\text{i,start}}) \log(1 - p_s^{\text{i,start}}) \\ & + y_i^{\text{i,end}} \log p_s^{\text{i,end}} + (1 - y_i^{\text{i,end}}) \log(1 - p_s^{\text{i,end}})) \end{aligned} \tag{10}$$

where $y_i^{i,start}$ represents the actual value of the $i\text{-}th$ token at the start of the head entity, and $y_i^{i,end}$ represents the actual value of the $i\text{-}th$ token at the end of the head entity.

The loss function for the tail entity tagging module as shown in (11):

$$\begin{aligned} L_{\text{tail}} = -\frac{1}{l}\sum_{i=1}^{l} & (y_i^{\text{i,start}} \log p_o^{\text{i,start}} + (1 - y_i^{\text{i,start}}) \log(1 - p_o^{\text{i,start}}) \\ & + y_i^{\text{i,end}} \log p_o^{\text{i,end}} + (1 - y_i^{\text{i,end}}) \log(1 - p_o^{\text{i,end}})) \end{aligned} \tag{11}$$

where $y_i^{i,start}$ represents the actual value of the $i\text{-}th$ token at the start position of the tail entity, $y_i^{i,end}$ represents the actual value of the $i\text{-}th$ token that is the end of the tail entity.

After training the two modules together, the joint loss function is represented as shown in (12):

$$L_{joint} = \alpha L_{head} + \beta L_{tail} \tag{12}$$

where α and β are adjustable hyper parameters.

V. Experiments

A. Datasets and evaluation metrics

Two publicly available datasets are used in our evaluation, DuIE2.0 [14] and CMeIE [15].

- DuIE2.0 [1] is the largest schema-based chinese relation extraction dataset in the field of relation extraction, which contains 48 predefined relation types.
- CMeIE [2] is a schema-based chinese medical information extraction dataset, which includes 53 well-defined relation types.

We divide both datasets into training set, validation set, and test set according to the ratio of approximately 8:1:1, and reorganize the relation categories. According to the principle of different overlapping mode relations, the sentences are divided into three categories: Normal, SEO and EPO. And according to the situation that the sentences contain different numbers of triples, they are divided into different sentences contain 1, 2, 3, 4 and ≥ 5 triples. Table I shows the statistics for the dataset.

TABLE I: Statistics of CMeIE and DuIE2.0 datasets.

	Datasets			Category							
	Train	Test	Val	Relation	Normal	SEO	EPO	N=1	N=2	N=3	N=4	$N \geq 5$
DuIE2.0	152,777	19,195	19,995	48	13,029	7,116	529	12,584	4,471	1,514	923	1,182
CMeIE	14,338	1,500	1,500	44	1,425	2,122	38	1,380	779	433	312	681

In this paper, the extraction performance of the model is judged according to whether the extracted triples (head, relation, tail) are correct. Following previous works [5], we also use standard micro Precision (P), Recall(R), and F1 score (F1) to measure the performances of our model.

B. Experimental settings

Since both datasets are in chinese, the encoder chooses the bert-base-chinese and chinese-bert-wwm-ext modules to compare. The overall model framework is PyTorch, and Adam [16] optimizer is used during the training phase. The learning rate of the encoder is set to 2e-5. The number of epochs is set to 50 and 100 for DuIE2.0 and CMeIE, respectively, and the batch size is set to 16. Also, to prevent overfitting, set the dropout value to 0.2. The weight parameters α and β in the equation (12) are set to 1.0 and 2.0, respectively. All hyperparameters are tuned on the validation set.

C. Baselines

In order to verify the effectiveness of the proposed model, we conduct comparative experiments with seven mainstream joint extraction models were proposed in recent years. Based on sequence-to-sequence models including CopyMTL [2] and WDec [17], based on table-filling models including TPLinker [13] and OneRel [18], based on tagging-based models including NovelTagging [19], CasRel [5], and RIFRE [20].

[1] https://aistudio.baidu.com/index

[2] https://tianchi.aliyun.com/dataset/95414

D. Main results

Table II shows the results of different joint extraction of the entities and relations models on DuIE2.0 and CMeIE. The encoder of all baselines uses BERT (chinese-bert-wwm-ext), and when running the comparison model, the hyper parameters refer to the description in the original paper, and the data settings and epoch numbers refer to the model in this paper.

TABLE II: Results on CMeIE and DuIE2.0 test sets. Bold marks stand for the best result of all models.

Method	DuIE2.0			CMeIE		
	$P(\%)$	$R(\%)$	$F1(\%)$	$P(\%)$	$R(\%)$	$F1(\%)$
NovelTagging	67.3	65.2	66.3	53.5	48.9	51.1
CopyMTL	49.6	39.4	43.9	31.7	26.8	29.1
WDec	64.1	54.2	58.7	29.1	18.4	22.5
CasRel	76.8	60.2	67.5	52.6	46.4	49.3
TPLinker	65.4	73.5	69.2	52.1	51.5	51.8
RIFRE	66.5	69.8	68.2	52.5	42.5	46.0
OneRel	71.7	66.2	68.3	46.3	38.1	41.8
Ours(bert-base-chinese)	76.7	60.4	67.6	51.3	50.7	51.0
Ours(chinese-bert-wwm-ext)	**77.7**	63.4	**69.8**	**53.7**	**53.0**	**53.3**

As can be seen from Table II, the $F1$ score of the model proposed in this paper on the CMeIE and DuIE2.0 is better than that of other models, which proves the effectiveness of the proposed model for the chinese relation triplet extraction task. Specifically, compared with models with redundant relations such as CasRel and RIFRE, the overall results of the proposed model are improved. However, the R still needs to be improved, and the analysis may be related to the entity threshold setting in this article. The performance of our model on the CMeIE still needs to be improved, which may be due to the strong specialization of CMeIE, which contains a large number of medical specialties terminology and medical-specific special symbols, which makes the model inadequately learn from the text.

Further, chinese-bert-wwm-ext has better extraction effect than bert-base-chinese because chinese-bert-wwm-ext supports integer word matching, two-way encoder and vocabulary masking, etc., which can better handle chinese semantic information and phrases. Experiment results demonstrate that the addition MHA module can enhance the model's attention to different features. And GAU can also enhance the feature expression ability of the original data and capture the features of the data distribution, thereby improving the generalization ability and performance of the model.

E. Analysis on different subtasks

In order to confirm that the model can handle complex scenarios, two additional experiments are carried out, as shown in fig. 2, from which it can be seen that the proposed model has achieved good results in the overlapping triples in both datasets. These results suggest that this paper is more conducive to the extraction of overlapping triples. Fig. 3 shows the results with different numbers of triples. As can be seen in fig. 3, our model better for different sentences contain 1, 2, 3, 4 and ≥ 5 triples. This shows that MHA and GAU components improve the performance of the model, and that the proposed method has the highest $F1$ score in all categories compared to baselines. This fully shows that the model proposed in this paper has a better ability to deal with the extraction of triple relations in complex scenes.

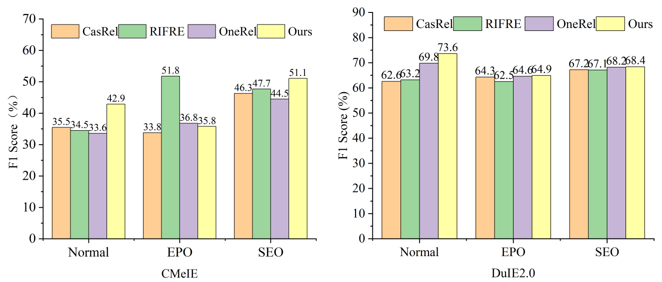


Fig. 2: According to the overlapping patterns in the partial match, different sentence types are obtained on the two datasets.

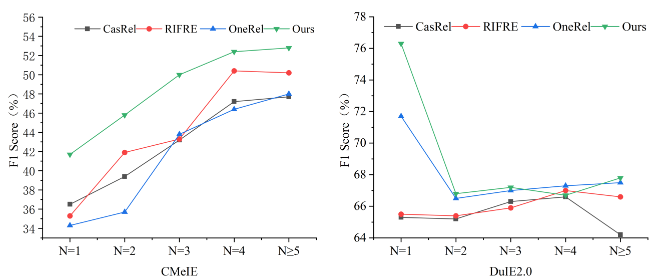


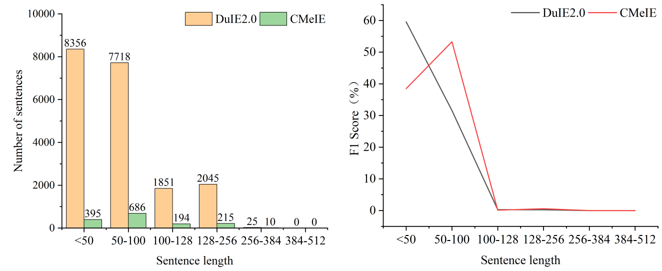
Fig. 3: Depending on the number of triples in the partial match, different sentence types are obtained on the two datasets.

F. Analysis on different sentence length

In order to analyze the impact of sentence length on the performance of the model, we divide sentence length into six subcategories, as shown in fig. 4 (a), the number of sentences in each subclass in the test set is statistically analyzed, and the extraction effect of the model in this paper under different sentence lengths is further compared, as shown in fig. 4 (b). As can be seen the model will affect the extraction effect when processing short or long sentences, and the extraction effect is better for shorter sentences with simple structure, so it is necessary to reasonably analyze the influence of sentence length on the accuracy of entity relation extraction.

G. Analysis of threshold changes

This section explores the impact of different entity thresholds on model performance. In entity threshold analysis, align the head-to-tail entity thresholds. Fig. 5 shows the changes in P, R, and $F1$ score at different threshold settings. As can be seen in fig. 5, with the increase in entity threshold, the P on the DuIE2.0 and CMeIE gradually increases, while the R decreases. When the thresholds are set to 0.5 and 0.4, the $F1$ score reaches its maximum on the DuIE2.0 and CMeIE. This



(a) Test set statistics. (b) $F1$ score on sentence lengths.

Fig. 4: Analysis on different sentence length.

change is due to the fact that different threshold settings have an impact on the number of detected relations and the number of triples extracted. In addition, it can be found that setting different entity thresholds on the two datasets has a certain degree of impact on the model performance.

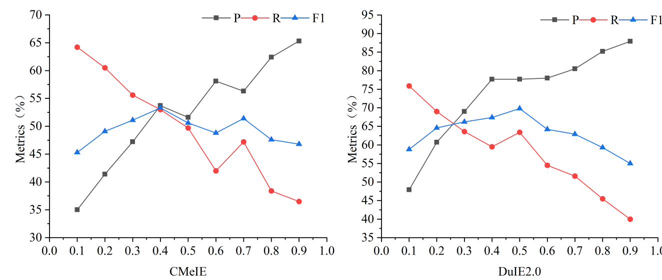


Fig. 5: The results of different entity thresholds.

H. Ablation study

To confirm the impact of each component on the overall performance of our model, we remove specific components from the original model and then conduct experiments. The ablation experiment results are reported in Table III.

TABLE III: Ablation study

	CMeIE			DuIE2.0		
	$P(\%)$	$R(\%)$	$F1(\%)$	$P(\%)$	$R(\%)$	$F1(\%)$
ALL	**53.7**	**53.0**	**53.3**	**77.7**	**63.4**	**69.8**
-head GAU	51.3	52.3	51.8	76.5	62.3	68.7
-head MHA	50.8	52.6	51.7	76.2	59.9	67.1
-tail GAU	47.8	52.8	50.2	75.9	59.5	66.4
-tail MHA	48.9	51.3	50.1	74.5	58.9	65.8

Specifically, when the GAU and MHA components of the head entity tagging module are removed, the $F1$ score of CMeIE and DuIE 2.0 are reduced, respectively. The reason for the decline in model performance is that the GAU component can enhance the semantic representation, and the MHA component allows the model to better capture key information in the input sequence, improve the model's understanding of the context, and help handle long-distance dependencies in the input sequence. And the additional features obtained by combining the two contain richer and deeper semantic information. Thus improving the joint extraction effect of the model on entity relations.

When the GAU and MHA components of the tail entity tagging module are deleted, the $F1$ score of CMeIE and DuIE2.0 also decreases, and the $F1$ score decreases more significantly than that when the GAU and MHA of the head entity is deleted, indicating that the fusion of relation vector h_r, subject vector v_s and token vector h_s is conducive to the understanding of complex contexts and the mining of global information. Moreover, MHA and GAU components enhance the classification capability of the relational classifier and improve the effect of the joint extraction method.

VI. Conclusions

In this paper, we propose a joint extraction of entities and relations based on pointer annotation, which effectively solves the overlapping triplet problem and reduces entity redundancy by applying different features to entities and relations, using the head entity tagging module to detect the subject, and then performing entity recognition to match different or identical tail entities. In addition, GAU and MHA components are designed to solve the problem of long-distance dependencies and enhance semantics. Experiment results demonstrate that our model is more effective than the baseline model, and can extract complex relation triples better, which can provide support for downstream tasks, including knowledge graph construction and knowledge question answering. However, there are still some areas where our model needs further improvement in future research. One is that the performance on the medical dataset CMeIE needs to be improved. The other is that the GAU and MHA have the same effect in the model in this paper, but they still need to be verified in other models.

Acknowledgment

This research is supported by the Natural Science Foundation of Inner Mongolia Autonomous Region (Grant No. 2023MS06023).

References

[1] X. Zeng, D. Zeng, S. He, K. Liu, and J. Zhao, “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)*, 2018, pp. 506–514.

[2] D. Zeng, H. Zhang, and Q. Liu, “Copymtl: Copy mechanism for joint extraction of entities and relations with multi-task learning,” in *Proceedings of the AAAI conference on artificial intelligence*, 2020, pp. 9507–9514.

[3] C. Gao, X. Zhang, H. Liu, W. Yun, and J. Jiang, “A joint extraction model of entities and relations based on relation decomposition,” *Int. J. Mach. Learn. Cybern.*, vol. 13, no. 7, pp. 1833–1845, 2022.

[4] J. Ning, Z. Yang, Y. Sun, Z. Wang, and H. Lin, “Od-rte: A one-stage object detection framework for relational triple extraction,” in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023, pp. 11 120–11 135.

[5] Z. Wei, J. Su, Y. Wang, Y. Tian, and Y. Chang, “A novel cascade binary tagging framework for relational triple extraction,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, D. Jurafsky, J. Chai, N. Schluter, and J. R. Tetreault, Eds. Association for Computational Linguistics, 2020, pp. 1476–1488.

[6] 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*. International Joint Conference on Artificial Intelligence, 2021.

[7] J. Chen, J. Hu, T. Li, F. Teng, and S. Du, “An effective relation-first detection model for relational triple extraction,” *Expert Systems with Applications*, vol. 238, p. 122007, 2024.

[8] A. Joshi, E. Fidalgo, E. Alegre, and L. Fernández-Robles, “Deepsumm: Exploiting topic models and sequence to sequence networks for extractive text summarization,” *Expert Systems with Applications*, vol. 211, p. 118442, 2023.

[9] P.-L. H. Cabot and R. Navigli, “Rebel: Relation extraction by end-to-end language generation,” in *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2021, pp. 2370–2381.

[10] Z. Yan, C. Zhang, J. Fu, Q. Zhang, and Z. Wei, “A partition filter network for joint entity and relation extraction,” in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*. Association for Computational Linguistics, 2021, pp. 185–197.

[11] F. Ren, L. Zhang, X. Zhao, S. Yin, S. Liu, and B. Li, “A simple but effective bidirectional framework for relational triple extraction,” in *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, 2022, pp. 824–832.

[12] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “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,NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds. Association for Computational Linguistics, 2019, pp. 4171–4186.

[13] Y. Wang, B. Yu, Y. Zhang, T. Liu, H. Zhu, and L. Sun, “Tplinker: Single-stage joint extraction of entities and relationsthrough token pair linking,” in *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13,2020*. International Committee on Computational Linguistics, 2020, pp. 1572–1582.

[14] S. Li, W. He, Y. Shi, W. Jiang, H. Liang, Y. Jiang, Y. Zhang, Y. Lyu, and Y. Zhu, “Duie: A large-scale chinese dataset for information extraction,” in *Natural Language Processing and Chinese Computing - 8th CCF International Conference, NLPCC 2019, Dunhuang, China, October 9-14, 2019, Proceedings,Part II*, ser. Lecture Notes in Computer Science, J. Tang, M. Kan, D. Zhao, S. Li, and H. Zan, Eds. Springer, 2019.

[15] T. Guan, H. Zan, X. Zhou, H. Xu, and K. Zhang, “Cmeie: Construction and evaluation of chinese medical information extraction dataset,” in *Natural Language Processing and Chinese Computing - 9th CCF InternationalConference, NLPCC 2020, Zhengzhou, China, October 14-18, 2020, Proceedings, Part I*, ser. Lecture Notes in Computer Science, X. Zhu, M. Zhang, Y. Hong, and R. He, Eds., vol. 12430. Springer, 2020, pp. 270–282.

[16] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015.

[17] T. Nayak and H. T. Ng, “Effective modeling of encoder-decoder architecture for joint entity and relation extraction,” in *Proceedings of the AAAI conference on artificial intelligence*, 2020, pp. 8528–8535.

[18] Y.-M. Shang, H. Huang, and X. Mao, “Onerel: Joint entity and relation extraction with one module in one step,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 36, 2022, pp. 11 285–11 293.

[19] S. Zheng, F. Wang, H. Bao, Y. Hao, P. Zhou, and B. Xu, “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, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, R. Barzilay and M. Kan, Eds. Association for Computational Linguistics, 2017, pp. 1227–1236.

[20] K. Zhao, H. Xu, Y. Cheng, X. Li, and K. Gao, “Representation iterative fusion based on heterogeneous graph neural network for joint entity and relation extraction,” *Knowledge-Based Systems*, vol. 219, p. 106888, 2021.