Entity Relation Joint Extraction Method Based on Insertion Transformers

Haotian Qi, Weiguang Liu, Fenghua Liu*, Weigang Zhu, Fangfang Shan
College of Computer, Zhongyuan University of Technology, Zhengzhou, Henan 451191, China

***Abstract*—Existing multi-module multi-step and multi-module single-step methods for entity relation joint extraction suffer from issues such as cascading errors and redundant mistakes. In contrast, the single-module single-step modeling approach effectively alleviates these limitations. However, the single-module single-step method still faces challenges when dealing with complex relation extraction tasks, such as excessive negative samples and long decoding times. To address these issues, this paper proposes an entity relation joint extraction method based on Insertion Transformers, which adopts the single-module single-step approach and integrates the newly proposed tagging strategy. This method iteratively identifies and inserts tags in the text, and then effectively reduces decoding time and the count of negative samples by leveraging attention mechanisms combined with contextual information, while also resolving the problem of entity overlap. Compared to the state-of-the-art models on two public datasets, this method achieves high F1 scores of 93.2% and 91.5%, respectively, demonstrating its efficiency in resolving entity overlap issues.**

***Keywords*—Entity relation extraction; tagging strategy; joint extraction; transformer**

I. Introduction

The proliferation of the Internet has led to an explosion of textual data, presenting a challenge in extracting valuable information efficiently. Various downstream tasks such as Knowledge Graph construction, Intelligent Question Answering Systems, and Recommendation Systems rely on the extraction of pertinent information from this unstructured textual data. Consequently, the extraction of entities and relations from text has emerged as a pivotal challenge in the field of Information Extraction. Entity relation extraction, as a core task within IE, aims to distil structured ternary information, namely <subject, relation, object> [1], from raw and unstructured text. This process is essential for furnishing crucial data support for subsequent tasks. Through accurate entity relation extraction, valuable insights can be gleaned from vast volumes of textual data, thereby enhancing the quality and efficiency of downstream applications.

Early entity relation extraction tasks typically employed a pipeline approach, which involved breaking down the extraction process into two distinct sub-tasks: Named Entity Recognition and Relation Extraction [2]. Initially, an entity recognition model would be constructed to identify entity pairs, followed by the development of a semantic relation model to perform relation extraction based on the recognized entity pairs. This sequential approach facilitated model construction but often resulted in issues such as data dependency, cascading errors, and information redundancy due to limited interaction between the tasks. In contrast, the joint extraction model integrates entity information and relations into a unified framework through joint training. This approach minimizes the drawbacks of the pipeline method by allowing for greater interaction between entity recognition and relation extraction. By simultaneously considering entity and relation information, the joint extraction model exhibits enhanced performance in handling diverse and complex semantic structures. Moreover, its parallel nature mitigates the accumulation of errors, thereby improving the overall efficiency and effectiveness of information extraction processes.

Depending on task complexity and design requirements, entity relation joint extraction can be classified into three fundamental architectures: multi-module multi-step, multi-module single-step, and single-module single-step [3]. The multi-module multi-step architecture showcases its modularity advantage in entity relation joint extraction. By segmenting tasks into multiple steps and modules, each module can concentrate on specific subtasks, thus enhancing flexibility and maintainability. However, this design can inadvertently propagate errors, diminishing overall performance and increasing training and optimization complexity. In contrast, the multi-module single-step architecture maintains modularity benefits while simplifying joint extraction. It reduces the risk of error propagation by sharing information across modules, making it suitable for handling relatively straightforward entity relation joint extraction tasks. However, it may sacrifice the capability to capture task complexity effectively. The single-module single-step architecture excels in its streamlined model structure, ease of training, and comprehensibility. However, since this approach decodes triples based on global information matrices, it may prolong decoding times and introduce issues such as excessive negative sampling.

As shown in Fig. 1, we use a rectangular solid to represent the number of computations in a single module and single-step process, with each block on the surface representing the computations between tokens under a single relationship. The red blocks indicate inefficient computations, while the green blocks represent efficient computations. Take the sample sentence “The dog got a driving license,” which contains an entity pair (The dog, owner, driving license). Assuming there are five pre-defined relationships and relationship r1 equals “owner”. From the figure, it is clear that this sample needs to be calculated $6 \times 6 \times 5$ times in total, but only 3 of those calculations are efficient. Hence, its reasoning efficiency is not high. When the sentence length increases, the number of inefficient calculations grows exponentially.

*Corresponding Author

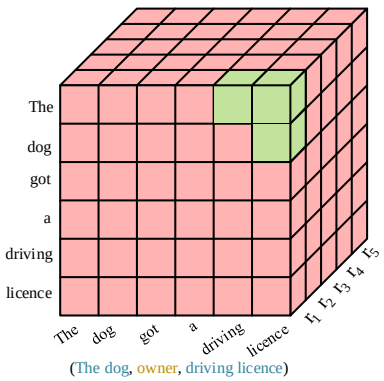

Fig. 1. Example of single-module single-step problem.

Addressing the challenges of time-consuming decoding and excessive negative sampling in current joint extraction processes, this paper proposes a single-module single-step approach based on Insertion Transformers. The method involves inserting a common token between each token of the sample during the inception stage. The model then assigns specific meanings to these tokens to recognize and delineate a triad. Subsequently, ordinary tokens are reintroduced in the vicinity of each identified special token, allowing the model to continue identifying and locating new triples following this logic until no more triples can be identified. In comparison to the baseline model, the proposed approach demonstrates notable improvements on both the NYT and WebNLG datasets. It not only significantly reduces decoding time but also effectively mitigates the number of negative samples compared to the state-of-the-art models. This further validates the performance of the proposed model.

The subsequent sections of this paper unfold as follows: Section II illustrates prior research endeavors. Section III details into the methodology employed. Section IV clarifies the experimental setup. Section V describes the experimental outcomes and engages in discussions. Lastly, Section VI summarizes the conclusions drawn from this study.

II. Related Works

This section provides an overview of the related work on entity relation joint extraction methods, where the single-module single-step based extraction method is the focus of this paper.

In recent years, with the advancement of deep learning, many scholars have integrated deep learning methods into models for entity relation extraction to enhance extraction accuracy. The majorities of entity relation extraction models in recent years have leveraged pre-trained language models, particularly BERT [4], and have exhibited remarkable performance. Among these, deep learning based on entity relation joint extraction can be categorized into three modelling approaches. There are early approaches based on global text information, including both multi-module multi-step and multi-module single-step methods. Besides, there is the single-module single-step method, which addresses the issue of cascading redundancy between modules and steps.

A. Multi-module Multi-step Method

The multi-module multi-step architecture offers the advantage of explicit task decomposition and modular design. To address the issue of ternary overlap in entity relation extraction tasks, Zeng et al. [5] introduced an end-to-end neural model called CopyRE, based on Seq2Seq (Sequence-to-sequence) with a Copy mechanism. This model can extract relational facts from sentences of different categories, including normal sentences, Single Entity Overlap (SEO), and Entity Pair Overlap (EPO) sentences. Subsequently, Zeng et al. [6] proposed the CopyMTL model, which adopts a multi-tasking framework and the Copy mechanism to predict multi-word entities. However, CopyRE struggles with extracting multi-word entities, and while CopyMTL improves this aspect, its effectiveness in extracting an arbitrary number of triples remains to be enhanced. To address this issue, Wei et al. [7] introduced the cascading Hierarchical Binary Tagging (HBI) model, CasRel, which conducts entity relation triple extraction in two successive steps. This model represents relations as a function mapping head entities to tail entities. Additionally, Tian et al. [8] proposed the HSL model, which employs a novel tagging scheme to convert the joint entity and relation extraction problem into a sequence tagging task using a hierarchical sequence tagging approach. Zheng et al. [9] proposed the PRGC model, which decomposes the task into three subtasks: relation judgment, entity extraction, and subject-object alignment. Geng et al. [10] proposed an attention mechanism integrating convolutional and recursive neural networks within a joint model, enhancing the utilization of contextual information. The FETI model, suggested by Chen et al. [11], integrates head-tail entity category information and employs an auxiliary loss function for more efficient utilization of entity category information. Ye et al. [12] introduced the CGT model, a ternary extraction model based on the generative Transformer, which leverages contrastive ternary-level calibration algorithms and batch-level dynamic attention masking mechanisms to enhance model performance. Yu et al. [13] optimized a joint extraction model for Chinese entity relation extraction using the RoBERTa pre-training model.

B. Multi-module Single-step Method

Compared to the aforementioned methods, the multi-module single-step simplifies the model structure and reduces extraction complexity. To address the issue of the model predicting the extraction order of multiple triples, Sui et al. [14] proposed an end-to-end network model, Set Prediction Networks (SPN), featuring Transformers-based features and non-autoregressive parallel decoding, along with a two-part matching loss. This model transforms the task of entity relation joint extraction into an ensemble prediction problem. Wang et al. [15] introduced a table-filling model, Table-Sequence Encoders, based on the Attention mechanism. This model facilitates the transfer and interaction of information between different input modalities by incorporating table encoders and sequence encoders. The TPLinker, proposed by Wang et al. [16], treats entity relation joint extraction as a tagged-pair linking problem and introduces a novel handshake tagging scheme for aligning entity-pair boundary tags under each relation type. Additionally, Wang et al. [17] proposed

UniRE, a table-filling model giving joint decoding, which employs a unified tag space and solves the problem of tag space dispersion in traditional entity relation extraction.

C. Single-module Single-step Method

The more lightweight single-module single-step approach simplifies extraction and enhances intuitiveness compared to the multi-module design. Kong et al. [18] introduced an end-to-end co-attention network called CARE, which utilizes a two-dimensional table to represent entity tags and relation tags respectively. Inspired by the modelling idea of Novel-Tagging [19], Shang et al. [3] proposed a fine-grained triple classification model, OneRel, at the entity token layer. This effectively mitigates issues such as cascading errors and entity redundancy.

Amalgamating the research on entity relation joint extraction, this paper proposes effective solutions to the shortcomings of existing methods in Section A. The primary contributions are as follows:

1) Integrating the concept of Insertion Transformers, we present a novel perspective by reframing the joint extraction of entity relations as fine-grained triple classification. Effectively reduce the decoding time of the model to a constant level, enhancing its efficiency by leveraging insertion operations.

2) A novel entity relation tagging strategy, Vanilla Entity Relation Tags (VRT), is proposed, significantly enhances the performance in addressing the issue of Entity Overlap.

3) We conduct model evaluations on two publicly available datasets, NYT and WebNLG. The results demonstrate that our approach surpasses current state-of-the-art baseline methods, particularly in handling the intricate context of overlapping triples.

D. Insertion Transformers

Insertion Transformers, a branch of non-autoregressive generative models originally proposed by Stern et al. [20], revolutionizes text generation. The core concept involves iteratively inserting elements into an initial blank sequence until the termination condition is met, effectively reducing inference time while maintaining high performance. Building upon this foundation, Gu et al. [21] introduced InDIGO, an insertion-based decoding algorithm that optimizes efficiency by reusing previous hidden states. Moreover, Zhang et al. [22] introduced POINTER, a hierarchical Transformer model that blends the strengths of BERT and Insertion Transformer, generating text through incremental token insertions. Additionally, the CBART model, proposed by He[23], enhances text generation by incorporating a token-level classifier at the encoder side. This classifier guides the decoder in performing substitution and insertion operations, enabling simultaneous fine-tuning of multiple input tokens and thereby enhancing the accuracy and efficiency of text generation.

III. Methodology

The objective of the single-module single-step model is to identify the set of triples present in a sentence for each relation. This involves identifying entity pairs within a given sentence, where the relation set of number M is represented as $R = \{r_1, r_2, \ldots, r_m\}$, and the sentence of length N is represented as $X = \{x_1, x_2, \ldots, x_n\}$. The task is to find the set of entity pairs $Y = \{(s_1, r_1, o_1), \ldots, (s_k, r_k, o_k)\}$ with conditional probability given by:

$$p(Y|X) = \prod_{m=1}^{M} \prod_{i=1}^{N} \prod_{j=1}^{N} p_\theta(Y_k|X, r_m) \quad (1)$$

where, θ represents the model parameter. However, the optimized single-module single-step model requires $N \times N$ computations to determine the entity pair set Y, which leads to exponential growth in computation for longer texts, significantly increasing decoding time. To address this issue, this method draws inspiration from the Insertion Transformers model, successfully reducing the number of decoding times to a constant level using insertion operations.

In this section, the specific implementation method will be discussed in five parts, A focuses on entity relation tagging strategy, B discusses decoding strategy of the model, and C will describe the overall model framework. D will delve into model training, finally E will explore the objective function used in this approach.

A. Entity Relation Tagging Strategy

In the task of entity relation joint extraction, the set of entity pairs in a sentence is unordered, and the positions of the head and tail entities in the sentence are also unordered. In order to enable the model to recognize the head and tail entities at any position in the text and extract the relation between the entities simultaneously, this approach proposes a novel entity relation tagging method. It consists of four special tags: Head Entity Relation Tag (HRT), Tail Entity Relation Tag (TRT), Overlap Entity Relation Tag (ORT), and a generic entity relation tag (Vanilla Entity Relation Tag, VRT). The definitions are as follows:

$$HRT = \{p_1, p_2, \ldots, p_M\} \quad (2)$$

$$TRT = \{p_{M+1}, p_{M+2}, \ldots, p_{2M}\} \quad (3)$$

$$ORT = \{p_{2M+1}, p_{2M+2}, \ldots, p_{3M}\} \quad (4)$$

$$VRT = p_0 \quad (5)$$

where, p indicates the token, and except for VRT, each entity relation token is assigned one by one corresponding to M relations.

As shown in Fig. 2, given the text $X = \{Jackie, \ldots, Island\}$, with $N = 7$. The set of entity pairs for this text is denoted as Y = {(*Jackie Chan*, *belong*, *Hong Kong*), (*Hong Kong*, *contain*, *Hong Kong Island*)}, with$K = 2$. The entity relation tagging method requires $K + 1$ steps of processing for this text. Except for the step 1, each step of processing can be divided into two operations, namely VRT insertion and VRT transformation. In the step 1, only VRT insertion operations are performed on the text X, and the insertion positions for VRT are between all adjacent tokens in the text, resulting in a total of $N + 1$ inserted p_0 tokens, yielding a new token sequence X'.

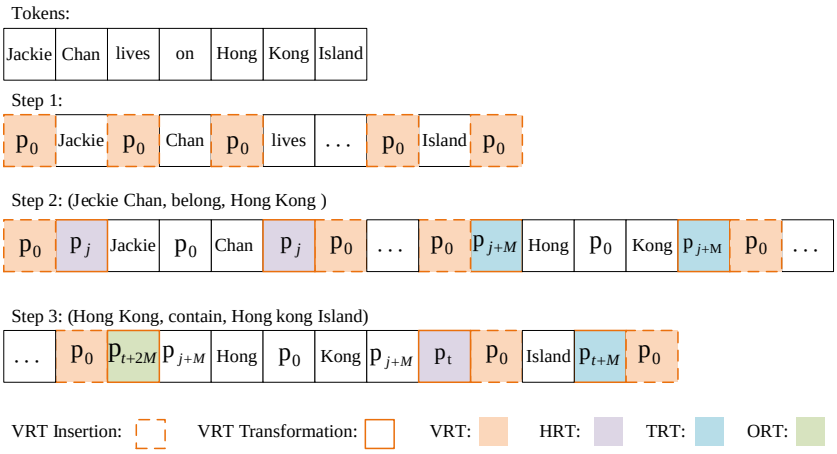

Fig. 2. Example of entity relation tagging.

Step 2 in Fig. 2 of the entity relation tagging method involves first performing VRT transformation on the text X', followed by VRT insertion operations. This sequence is beneficial for generating subsequent training samples and targets. In VRT transformation, VRTs are converted to other tags to enclose the entity pairs in the sentence. Specifically, if '*belong*' is at the j position in the relation set R, then p_j is taken as HRT and p_{j+M} as TRT. In text X, VRT tags preceding the first token of the head entity and following the last token are converted to HRT. Similarly, VRT tags preceding the first token of the tail entity and following the last token are converted to TRT. Finally, VRTs are inserted around the newly converted HRT and TRT.

The operation of step 3 is the same as step 2 in Fig. 2. However, due to the overlapping entities in (*Hong Kong*, *contain*, *Hong Kong Island*), there are some differences in VRT transformation. Specifically, VRTs preceding '*Hong*' need to be converted to both HRT and TRT. In this case, the ORT is replaced with p_{t+2M} for this position, where t is the index of 'contain' in the relation set R.

Entity overlapping addressed in the step 3 constitutes one facet of the broader entity overlapping challenges, and this method posits that the proposed entity relation tagging method remains efficacious in effectively mitigating such issues. Specifically, the entity overlap conundrum can be delineated into three distinct categories: EPO, HTO, and SEO, where SEO encapsulates instances of overlapping entities within triplet sets, encompassing both EPO and HTO scenarios.

Focusing solely on a singular entity pair during each processing step, the entity relation tagging method sidesteps occurrences where entities overlap within other entity pairs, thereby aptly resolving the quandary of individual entities overlapping with other entity pairs in both EPO and SEO settings. Moreover, given the inherent significance of entity positioning entailed by the tagging position within the entity relation tagging method, the spatial arrangement of entities within the sentence exerts an influential impact on the methodology, beyond merely addressing the HTO challenge. To redress this issue, it becomes imperative to ensure the uniqueness of encoding. As shown in Fig. 3, the method categorizes the distribution of entities within sentences into three types: non-overlapping head and tail entities, partially overlapping head and tail entities, and completely overlapping head and tail entities. Among them, partially overlapping head and tail entities can be further subdivided into five scenarios.

Within this framework, “HB” signifies the head token of the head entity, “HE” denotes the tail token, “TB” designates the head token of the tail entity, and “TE” indicates the tail token. The corresponding encoded values depicted on the right-hand side of the illustration affirm the distinctiveness of encoding across all scenarios, thus ensuring the integrity of the decoding process devoid of errors or decoding failures.

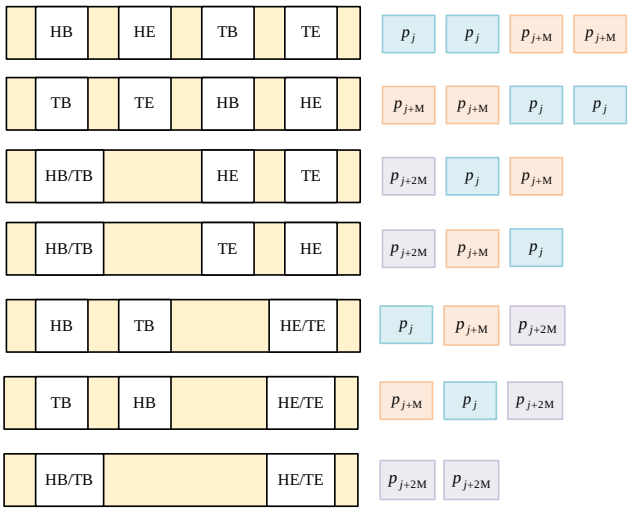

Fig. 3. Distribution of entity relation tagging.

B. Decoding Strategy

This training strategy of the method involves predicting each VRT for every sample, determining its corresponding entity relation tagging. For a sentence containing K triples, the entity relation tagging method conducts $K+1$ steps of preprocessing on the sentence. The essence of this training

method lies in treating the VRT insertion results of each step i (where $i \geq 1$ and $i < k$) as training samples and the VRT transformation results of step $i+1$ as the corresponding training targets. When $i = k+1$, the VRT insertion results serve both as training samples and targets. Essentially, this approach involves performing multi-class classification prediction on the VRT matrix of length N. This entails jointly inputting the VRT matrix and the text into the model and predicting the entity relation tag each VRT matrix output corresponds to.

The essence of this training method lies in conducting multi-class classification predictions on VRT matrices of length N. This involves inputting the VRT matrix and text into the model simultaneously, and predicting the entity relation tagging on the output of the VRT matrix.

Decoding strategy outlined in this approach involves a stepwise process. Initially, the first step of entity relation tagging is taken as input, and then fed into the model to obtain an entity pair. Subsequently, this entity pair is tagged, and the process iterates by continually feeding it back into the model to obtain the next entity pair. This iterative process continues until the model's output no longer yields a complete entity pair.

Entity relation tagging leverages information about the position of the tag, as well as distinct head, tail, and overlap tags, to ascertain the head and tail entities within a sentence. Furthermore, the relation between the head and tail entities is determined based on the relation information encoded in the tokens. By multi-stage learning, the model becomes proficient in mapping the VRT matrix to individual entity pair tagging. This approach offers several advantages:

1) The VRT matrix maintains an appropriate level of sparsity, facilitating simple and efficient tagging predictions by the model.

2) It reduces computational overhead, as computing all entity pairs for a given sample typically only requires about K iterations.

3) The processing of model utterances becomes more comprehensive and holistic, enhancing its overall effectiveness.

C. Model Framework

Our method focuses on single-module single-step entity relation extraction, emphasizing the refinement achieved through methods like multi-head attention mechanisms and cross-attention mechanisms to gradually extract entity pairs and their associated relations from textual data. Employing an innovative tagging approach, we integrate entity recognition and relation extraction into a unified process, enabling the system to understand the information in the text at different levels and thus better extract entities and relations.

1) Multi-head attention mechanism: Multi-Head Attention Mechanism is an enhanced technique derived from the self-attention mechanism. Self-attention is a method capable of determining the importance of each position in the input sequence, thus effectively addressing long-distance dependencies within the sequence. In the context of joint extraction of entity relations, diverse aspects may need to be considered simultaneously, necessitating the use of multiple self-attention mechanisms to handle these varied concerns. Multi-Head Attention involves employing multiple self-attention mechanisms on an input sequence to obtain several sets of attention results. Subsequently, these results are concatenated and linearly projected to yield the final output.

$$MultiHead(Q,K,V) = Concat(head_1, \cdots head_n)W^O \quad (6)$$

$$head_i = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right) \quad (7)$$

where, $W_i^Q \in \mathbb{R}^{d_m \times d_k}$, $W_i^K \in \mathbb{R}^{d_m \times d_k}$, $W_i^V \in \mathbb{R}^{d_m \times d_v}$, and $W_i^O \in \mathbb{R}^{nd_v \times d_m}$ are the projection matrices learned by the model based on the text after adding the VRT, with O as the output tensor, where q, k, v represent the dimensions of the query, key, and value vectors, respectively.

2) Cross-attention mechanism: Cross-attention mechanism is a special form of multi-head attention that splits the input tensor into two parts $X_1 \in \mathbb{R}^{n \times d_1}$ and $X_2 \in \mathbb{R}^{n \times d_2}$, and then uses one of the parts as a query set and the other as a key-value set. Its output is a query of size $n \times d_2$ tensor, and for each row vector, its attentional weight for all row vectors is given.

Specifically, let $Q = X_1 W^Q$, $K = V = X_2 W^K$, then the cross-attention is calculated as follows:

$$CrossAttention(X_1, X_2) = Soft\max\left(\frac{QK^T}{\sqrt{d_2}}\right)V \quad (8)$$

where, $W^Q \in \mathbb{R}^{d_1 \times d_k}$, $W^K \in \mathbb{R}^{d_2 \times d_k}$ are the projection matrices learned by the model based on the text after adding the VRT, and d_k is the dimension of the key-value set and also the dimension of the query set.

D. Training Strategy

As illustrated in Fig. 4, this approach utilizes BERT [4] as the encoder to initially encode the text sequence. On the decoder side, the input involves inserting the VRT code or other entity relation marker codes into the text sequence code. Initially, the decoder conducts self-attention on the text sequence with inserted entity relation markers, thereby allowing the markers to acquire coarse-grained contextual information. Given that the entity relation markers disrupt the original text sequence, acquiring complete sequence information becomes challenging. Hence, we introduce a cross-attention mechanism at the decoder's backend, enabling the input sequence to focus on the entire text sequence. Following the output, a masking operation is performed on the non-VRT tokens to solely obtain the output of the VRT tokens. The resulting probability of the target is determined as:

$$p(Y|X,u,t;\theta) = \prod_{i=1}^{n+1} p(y_i|X,u_i,t;\theta) \quad (9)$$

where, X represents the text sequence, u denotes the VRT marker, t signifies the HRT, TRT, or ORT marker alongside the VRT marker, and θ represents the model parameter. n denotes the length of the text sequence, and since the VRT marker adds one unit to the length of the text sequence, it is represented as $n+1$.

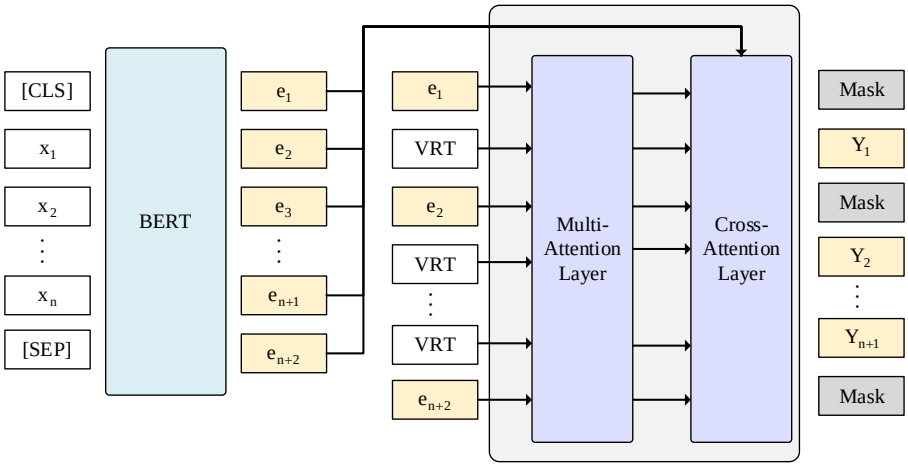

Fig. 4. Entity relation joint extraction framework based on insertion transformers.

E. Objective Function

The objective function of the model in this approach is defined as follows:

$$L = -\frac{1}{n+1}\sum_{i=1}^{n+1} log\ p(y_i|X, u_i, t; \theta) \quad (10)$$

IV. Experimental Setup

In this section, we first introduced the datasets and evaluation criteria used in the experiments, providing a detailed breakdown of the composition of the two datasets, and outlined the experimental details.

A. Datasets and Evaluation Metrics

1) Datasets: To provide a more robust explanation of the results of this model, two widely used datasets in relation and entity extraction tasks were adopted in this study: the New York Times (NYT) dataset [24] and the WebNLG dataset [25]. The details of the dataset sources and divisions are outlined below:

NYT is a renowned dataset for distant supervision relation extraction tasks. It utilizes Freebase online database, which stores entities and their relations, as the distant supervision source. It consists of articles from The New York Times annotated with named entity tags, coreference chains, and relation mentions. It contains approximately 1.8 million articles.

WebNLG consists of triple sets describing entities and their relations in natural language text. Initially used for natural language generation challenges, it later became the most commonly used general-domain dataset for evaluating triple extraction models, comprising data converted from the DBpedia knowledge base into natural language text. It consists of around 25,000 English sentences paired with RDF triples, offering a diverse range of content for text generation tasks. Detailed data are shown in Table I.

This approach conducted statistical analysis on the training, validation, and test sets of both datasets. Additionally, we categorized the datasets based on four different types of triple overlap patterns.

TABLE I. The Statistics of NYT and WebNLG.

Category	NYT		WebNLG	
	Train	**Test**	**Train**	**Test**
Normal	37013	3266	1596	246
EPO	9782	978	227	26
SEO	14735	1297	3406	457
Total	56195	5000	5019	703

Both the NYT and WebNLG datasets come in two versions: one version annotates solely the final word of entities, while the other annotates the entire span of entities. We denote the first version datasets as NYT* and WebNLG*, and the second version as NYT and WebNLG, respectively.

2) Evaluation metrics: To comprehensively evaluate system performance, we adopt three fundamental evaluation metrics consistent with the field of entity relation joint extraction: precision (P), recall (R), and the harmonic mean F_1 measure, for comparison with other baseline models. Their formulas are as follows:

$$precision = \frac{TP}{TP+FP} \quad (11)$$

$$recall = \frac{TP}{TP+FN} \quad (12)$$

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (13)$$

where, TP, FP, and FN represent true positives, false positives, and false negatives, respectively. In our experiments, correctness or incorrectness is considered with respect to triplets. That is, a triplet result is considered correct only when h_1(head entity), r(relation), and t_1(tail entity) are predicted correctly.

B. Experimental Environmental Details

The experiments were conducted on an Ubuntu 20.04 LTS operating system, utilizing hardware comprising an NVIDIA RTX-A2000-GPU with 12 GB of memory, an Intel i7-10700K CPU, and 64 GB of RAM. The software stack included Python version 3.7, PyTorch version 1.7 for deep learning frameworks, and CUDA version 11.4. The NVIDIA driver version was

470.94, and the pre-trained model library used was Transformers version 4.6.1. The experiments employed an AdamW optimizer.

V. Results and Discussion

This section assesses the effectiveness of our data by comparing experimental outcomes with those of other baseline models. We utilize commonly employed general-domain datasets and evaluation metrics for evaluating triplet extraction models. Moreover, we present experimental results across different datasets using OneRel [3] as a comparative method to validate our experiments. Additionally, we analyze and interpret precision, recall, and F1-score. Besides, we utilize ablation experiments to ascertain whether the cross-attention mechanism in our experiments plays a crucial role in controlling the model's triplet extraction. Finally, we compare the inference times of the models to confirm our model's high inference efficiency.

In order to evaluate the impact of the proposed joint extraction method based on Insertion Transformers on text, Table 2 in this section presents the experimental results of this method and other approaches on the NYT and WebNLG, with best results are highlighted in bold. The experimental results for the contrastive models SPN [14], CasRel [7], TPLinker [16], PRGC [9], OneRel [3], and CGT [12] are sourced from[26]. By comparing metrics such as F1 score, our method shows improvement over baseline models like OneRel in the task of entity relation joint extraction.

A. Experimental Results and Analysis

1) Experimental results: Table II presents a comparative analysis of the proposed model and baseline models across the NYT*, WebNLG*, NYT, and WebNLG datasets.

TABLE II. Comparison Results of Joint Model Performance Based on NYT and WebNLG: %

Method	Model	Partial annotation						Full annotation					
		NYT*			WebNLG*			NYT			WebNLG		
		P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Multi-module multi-step	CasRel [7]	89.7	89.5	89.6	93.4	90.1	91.8	-	-	-	-	-	-
	PRGC [9]	93.3	91.9	92.6	94.0	92.1	93.0	93.5	91.9	92.7	89.9	87.2	88.5
	CGT [12]	**94.7**	84.2	89.1	92.9	75.6	93.4	-	-	-	-	-	-
Multi-module single-step	TPLinker [16]	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 [14]	93.3	91.7	92.5	93.1	93.6	**93.4**	92.5	92.2	92.3	-	-	-
Single-module single-step	OneRel [3]	92.8	92.9	92.8	**94.1**	**94.4**	94.3	93.2	92.6	92.9	91.8	90.3	91.0
	Ours	94.2	**93.4**	**93.0**	93.6	93.7	93.2	**94.8**	**93.8**	**93.2**	**92.0**	**91.2**	**91.5**

On the NYT* dataset, our model generally exhibits superior recall and F1 scores compared to baseline models, with precision slightly lower than the CGT model by 0.5 but still ahead of other models. This might be attributed to the fact that the number of negative samples in our training set exceeds that of CGT, although the accuracy is slightly lower than CGT, it exhibits superior performance in terms of recall and F1 score. On the NYT dataset, our model has achieved comprehensive superiority, with F1 scores outperforming PRGC, TPLinker, and OneRel by 0.5, 1.9, and 0.3 percentage points, respectively. Similarly, on the WebNLG dataset, the F1 scores are correspondingly higher by 3, 4.8, and 0.5 percentage points. On the WebNLG* dataset, our model slightly lags behind OneRel but outperforms other models. This may be because, compared to entities with only one token in WebNLG*, our model is better adapted to learning entities with multiple tokens in WebNLG. Meanwhile, on the WebNLG dataset, our model surpasses OneRel in precision, recall, and F1 scores. This indicates that the proposed method is better suited for fully annotated data, as entity relation tagging requires contextual information, and annotating more context is beneficial for subsequent predictions.

2) Results discussion and analysis: Among the multi-module multi-step baseline models, CGT exhibits slightly higher precision than our model on the NYT* dataset, while PRGC shows slightly higher precision on the WebNLG* dataset. However, our model consistently outperforms CGT, PRGC, and CasRel in precision, recall, and F1 scores, owing to the single-module's capability to handle text and relations efficiently, resulting in fewer errors compared to multi-module approaches.

Among the multi-module single-step baseline models, our model consistently outperforms them by 1 to 2 percentage points. This is attributed to the adoption of a joint decoding algorithm in the multi-module single-step approach, which reduces cascading errors compared to the multi-module multi-step approach. However, there are still some redundant errors when combining triplets, indicating limitations. In contrast, our proposed method extracts a complete entity pair in a single pass based on the text sequence, reducing redundant errors.

Compared to the single-module single-step model OneRel, our model demonstrates varying degrees of superiority on the NYT*, NYT, and WebNLG datasets. OneRel exhibits fewer errors compared to other models. However, due to the interference from lengthy text, OneRel predicts all possible entity pairs, leading to a decrease in prediction accuracy. This results in our model outperforming OneRel in terms of accuracy. Overall, our proposed method surpasses baseline models.

Besides, validating our method capability in addressing overlapping patterns and handling multiple triples, we conduct two additional experiments on distinct subsets of NYT* and WebNLG*. We employ five robust models as baseline comparators, and the comprehensive results are presented in Table III.

TABLE IV. F1-SCORE ON SENTENCES WITH DIFFERENT TRIPLE NUMBERS. ON NYT* AND WEBNLG*: %

Model	NYT*					WebNLG*				
	N=1	N=2	N=3	N=4	N≥5	N=1	N=2	N=3	N=4	N≥5
CasRel ¶	88.2	90.3	91.9	94.2	83.7	89.3	90.8	94.2	92.4	90.9
PRGC ¶	**91.1**	93.0	93.5	95.5	93.0	89.9	91.6	95.0	94.8	92.8
TPLinker ¶	90.0	92.8	93.1	96.1	90.0	88.0	90.1	94.6	93.3	91.6
SPN ¶	90.9	93.4	94.2	95.5	90.6	89.5	91.3	**96.4**	94.7	93.8
OneRel ¶	90.5	93.4	93.9	**96.5**	94.2	91.4	93.0	95.9	**95.7**	94.5
Ours	91.0	**93.6**	**94.5**	96.3	**94.4**	**92.1**	**93.3**	95.6	94.4	**94.7**

[a.] Mark ¶ Indicates Results from [3]. "N" means different triple numbers, with the sentences were categorized from the test sets into five subclasses. Each class includes sentences that consist of 1, 2, 3, 4, or >5 triples.

It can be observed that our model achieves the best F1 scores in six out of ten categories, especially in the case of $N \geq 5$. Sentences with $N \geq 5$ may simultaneously contain Normal, SEO, EPO, and HTO patterns, making the extraction more complex. Importantly, our method performs best on $N \geq 5$ for both NYT and WebNLG*, demonstrating the effectiveness of our VRT tagging in addressing overlapping triplets from the model design perspective. This validates the efficacy of our model.

In models guided by a multi-module multi-step approach, it is demonstrated that sufficient interaction between head and tail entity information and relation information positively impacts model performance. Subsequently, employing a multi-module single-step model shows improved expressive power, indicating that using joint decoding instead of independent decoding in multiple steps helps alleviate cascading errors between steps and thus enhances model performance.

As for single-module single-step methods, although studies related to this are relatively scarce, from a performance perspective, integrating multiple sub-modules also reduces cascading errors between modules, leading to performance improvement. Considering the performance of joint models on the NYT and WebNLG datasets, the modeling direction of entity relation joint extraction is moving towards the idealized single-module single-step modeling method.

B. Ablation Study

Ablation experiments were conducted to investigate whether the cross-attention mechanism of the approach benefits the prediction of entity relation tagging. To assess the importance of the cross-attention mechanism, two experiments were conducted separately on the NYT and WebNLG datasets, one without cross-attention and the other with cross-attention. The keyword tagged with # denote those without using cross-attention. The experimental results are shown in Table IV.

It can be observed that after using the cross-attention mechanism, the accuracy in predicting entity pairs significantly improved. This indicates that after the VRT markers undergo cross-attention, they obtain more complete text information, enhancing the model's ability to predict entity relation tagging.

TABLE V. ABLATION EXPERIMENT ON NYT AND WEBNLG: %

Dataset	Evaluation	Ours	#Ours
NYT	**Precision**	93.8	88.2
	Recall	93.0	87.8
	F1	93.2	87.3
WebNLG	**Precision**	92.0	87.9
	Recall	91.2	86.2
	F1	91.5	86.1

C. Model Efficiency Analysis

The objective of this section is to evaluate model efficiency through the measurement of inference time, with a maximum sequence length set to 128. Sentence lengths are segmented into four intervals: (0, 32], (32, 64], (64, 96], and (96, 128]. The purpose of this segmentation is to precisely investigate the impact of sequence length variation on model performance.

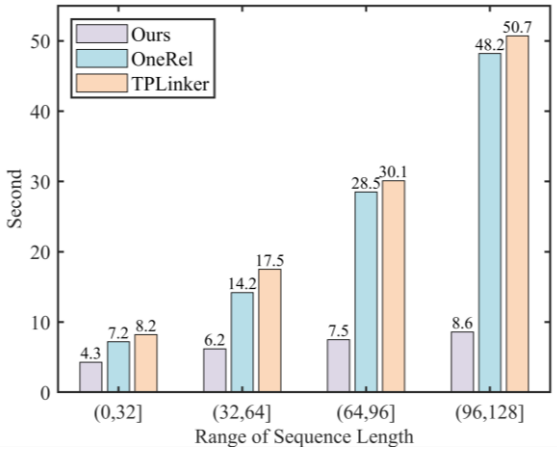

Fig. 5. Comparison of inference time.

As revealed by the results in Fig. 5, with an increase in sequence length, the inference times of OneRel and TPLinker models noticeably accelerate. This is due to their requirement to process all possible combinations between each pair of tokens across different relational contexts, resulting in an exponential increase in inference time as sequence length increases. In contrast, our model's inference time is significantly lower than the comparative models,

with a more gradual growth trend, as it predominantly relates to the number of entity pairs in the sequence, and there is a gentle positive correlation between the number of entity pairs and the increase in sequence length.

VI. CONCLUSION

In this paper, we utilize the Insertion Transformers framework to refine the task of entity relation joint extraction, introducing a novel VRT tagging strategy. This approach allows for more precise capturing of entity relation triplets, effectively addressing issues related to entity overlap in triplets and significantly reduces the inference time, thereby enhancing the efficiency of the model. Experimental evaluations on two public datasets demonstrate the superior performance of our model compared to state-of-the-art models across different scenarios.

In the future, we plan to delve into the following directions: we aim to devise a more efficient VRT tagging strategy to further enhance its ability to capture the associations between entities and relations, thereby making the model more efficient and focused. We also intend to investigate the concept of triplet overlap in other information extraction tasks, such as event extraction.

ACKNOWLEDGMENT

Project National Natural Science Foundation China (No: 62302540).

REFERENCES

[1] G. Zhou,J. Su, J. Zhang, “Exploring various knowledge in relation extraction,” In Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, pp. 427-434, 2005.

[2] D. Zeng, K. Liu, S. Lai, “Relation classification via convolutional deep neural network,” In Proceedings of the 25th International Conference on Computational Linguistics, pp. 2335-2344, 2014.

[3] Y. Shang, H. Huang, 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, no. 10, pp. 11285-11293, 2022.

[4] J. Devlin , M. Chang , K. Lee, “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, pp. 4171-4186, 2019.

[5] X. Zeng, D. Zeng, S. He, “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, pp. 506-514, 2018.

[6] D. Zeng, H. Zhang, 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, vol. 34, no. 05, pp. 9507-9514, 2020.

[7] Z. Wei, J. Su, Y. Wang, “A novel cascade binary tagging framework for relational triple extraction,” In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 1476-1488, 2020.

[8] J. Tian, X. Lv, X. You, “Hierarchical Sequence Annotation Based Joint Extraction Method for Entity Relation,” Journal of Peking University (Natural Science Edition), vol. 57, no. 1, pp. 53-60, 2021.

[9] H. Zheng, R. Wen, X. Chen, “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, pp. 6225-6235, 2021.

[10] Z. Geng, Y. Zhang, Y. Han, “Joint entity and relation extraction model based on rich semantics,” Neurocomputing, pp. 132-140, 2020.

[11] R. Chen, X. Zheng, Y. Zhu, “Joint entity and relation extraction fusing entity type information,” Computer Engineering, vol. 48, no. 3, pp. 46-53, 2022.

[12] H. Ye, N. Zhang, S. Deng, “Contrastive triple extraction with generative transformer,” In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 16, pp. 14257-14265, 2021.

[13] K. Yu, F. Huang, Q. Wu, “Joint extraction method for Chinese entity relation based on bidirectional semantics,” Computer Engineering, vol. 49, no. 1, pp. 92-99, 2023.

[14] D. Sui, Y. Chen, K. Liu, “Joint entity and relation extraction with set prediction networks,” IEEE Transactions on Neural Networks and Learning Systems, 2023.

[15] J. Wang, W. Lu, “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, pp. 1706-1721, 2020.

[16] Y. Wang, B. Yu, Y. Zhang, “TPLinker: Single-stage joint extraction of entities and relations through token pair linking,” In Proceedings of the 28th International Conference on Computational Linguistics, pp. 1572-1582, 2020.

[17] Y. Wang, C. Sun, Y. Wu, “UniRE: A unified tag 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, pp. 220-231, 2021.

[18] W. Kong, Y. Xia, “CARE: Co-Attention Network for Joint Entity and Relation Extraction,” arXiv preprint arXiv:2308.12531, 2023.

[19] S. Zheng, F. Wang, H. Bao, “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, pp. 1227-1236, 2017.

[20] M. Stern, C. William, K. Jamie, U. Jakob, “Insertion transformer: Flexible sequence generation via insertion operations,” In International Conference on Machine Learning, pp. 5976-5985, 2019.

[21] J. Gu, L. Qi, C. Kyunghyun, “Insertion-based decoding with automatically inferred generation order,” Transactions of the Association for Computational Linguistics. pp. 661-676, 2019.

[22] Y. Zhang, G. Wang, C. Li, Z. Gan, C. Brockett, & W. Dolan, “POINTER: Constrained Progressive Text Generation via Insertion-based Generative Pre-training,” In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing. pp. 8649-8670, 2020.

[23] X. He, “Parallel Refinements for Lexically Constrained Text Generation with BART,” In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 8653-8666, 2021.

[24] S. Riedel, L. Yao, & A. McCallum, “Modeling relations and their mentions without tagged text,” In Machine Learning and Knowledge Discovery in Databases: European Conference, pp. 148-163, 2010.

[25] C. Gardent, A. Shimorina, S. Narayan, & L. Perez, “Creating training corpora for NLG micro-planning,” In 55th Annual Meeting of the Association for Computational Linguistics, pp. 179-188, 2017.

[26] Y. Zhang, S. Liu, Y. Liu, “A review of deep learning-based entity-relation joint extraction research,” Journal of Electronics, pp. 1093-1116, 2023.