A Boundary Token Graph for Zero-Shot Relation Triplet Extraction Involving Discontinuous Entities

Kailun Lyu[*], **Zehan Li**[*], **Fu Zhang**[†], **Jingwei Cheng**
School of Computer Science and Engineering, Northeastern University, China
lyukailun@163.com, lizehan1999@163.com, zhangfu@neu.edu.cn, chengjingwei@neu.edu.cn

Abstract

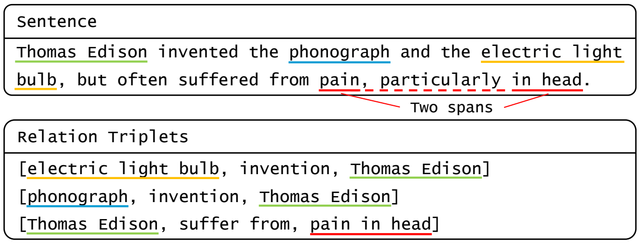
Zero-Shot Relation Triplet Extraction (ZSRTE) aims to extract head-tail entity pairs and their corresponding relations from sentences, where the relations available during inference are not seen during training. Existing methods typically assume that entities are continuous; however, in practice, entities can be discontinuous, which poses challenges to these approaches. To address this issue, we are the first to discuss and study the ZSRTE task involving discontinuous entities, and propose an innovative **BoG** framework, which is based on our proposed **Boundary Token Graph** structure. This method first predicts and adds edges between boundary tokens of (dis)continuous entities to construct a token graph, and then innovatively transforms the relation triplet extraction task into a process of finding paths in the graph. Additionally, we design a Boundary Token-Aware Prompt for each relation to further enhance the interaction between boundary tokens and relation semantics. Experimental results on four ZSRTE datasets—with or without discontinuous entities—consistently demonstrate that our method outperforms previous approaches, achieving state-of-the-art results.

Code and Datasets — https://github.com/LyuKaiLun/BoG

Introduction

Relation Triplet Extraction (RTE) involves extracting relation triplets in the format `[Head Entity, Relation, Tail Entity]` from unstructured text. Since existing methods (Zheng et al. 2017; Wang and Lu 2020) often rely heavily on large amounts of annotated data, Chia et al. (2022) introduced the Zero-Shot Relation Triplet Extraction (ZSRTE) task to mitigate this dependency. In this task, the candidate relations during inference are unseen, which means they are not present in the training set.

The head and tail entities, as key components of a relation triplet, are not always continuous in some more realistic scenarios or in some specific domains (Xia et al. 2023; Corro 2024). A discontinuous entity consist of multiple non-contiguous spans rather than a single span. As shown in **Figure 1**, consider the string "*pain, particularly in head*", where entity "*pain in head*" is discontinuous, with a gap involving intervening words.


Sentence

Thomas Edison invented the phonograph and the electric light bulb, but often suffered from pain, particularly in head.

Relation Triplets

[electric light bulb, invention, Thomas Edison]
[phonograph, invention, Thomas Edison]
[Thomas Edison, suffer from, pain in head]

Figure 1: An example of a relation triplet involving a discontinuous entity, where the entity "*pain in head*" consists of two spans: "*pain*" and "*in head*".

Research on discontinuous entities has primarily focused on the domain of Named Entity Recognition (NER). Most approaches recognize discontinuous entities by either extending the traditional BIO tagging scheme (Tang et al. 2013; Xu et al. 2015; Metke-Jimenez and Karimi 2016; Dai, Karimi, and Paris 2017; Tang et al. 2018) or leveraging specific structural models (Dai et al. 2020; Wang et al. 2021a). However, these methods primarily focus on identifying entities in isolation, without taking into account the relationships between entities. As a result, they are ill-suited for tasks like relation triplet extraction, which requires not only identifying entities but also capturing intricate interactions between entities.

In the field of ZSRTE, existing **discriminative methods** (Lv et al. 2023; Gong and Eldardiry 2024; Lan et al. 2024) primarily identify entities by predicting their left and right positions within a sentence. Nonetheless, these methods **assume that all entities are continuous**, making them incapable of handling discontinuous entities. In contrast, recent **generative methods** (Chia et al. 2022; Kim et al. 2023; Xu et al. 2024) utilize Sequence-to-Sequence (Seq2Seq) models to generate entities in textual form. While these generative methods can extract triplets involving discontinuous entities, they are inherently insensitive to discontinuity during the decoding process. Their underlying assumption that entities are continuous by design leads to a lack of explicit modeling for discontinuous entities, particularly when large gaps or intervening distractor words exist between spans, which

---

[*] These authors contributed equally.
[†] Corresponding author.

often results in inaccuracies in boundary recognition.

To address these challenges, we propose a discriminative ZSRTE method for extracting relation triplets by fine-grained identification of the boundaries for discontinuous entities. To achieve this, we introduce the innovative **BoG** framework, which is based on our proposed **Boundary Token Graph** structure. In this approach, the graph's vertices represent the left or right boundary tokens of each span. Once edges (including span-intra edges and span-inter edges) between vertices are predicted, each path in the directed graph corresponds to a distinct triplet, thereby innovatively transforming the ZSRTE task into the problem of finding paths within this graph. Further, to enhance the model's understanding of relations, we manually design a **Boundary Token-Aware Prompt** for each relation. The prompt includes the relation description and learnable **Boundary Token Markers**, which help distinguish whether a boundary token belongs to the left or right boundary of a span, thereby strengthening the interaction between boundary tokens and relation semantics.

Our contributions are summarized as follows:

- To the best of our knowledge, we are the first to focus on the relation triplet extraction involving discontinuous entities under the zero-shot setting.
- We propose an innovative Boundary Token Graph structure, which enables the extraction of all relation triplets by finding all paths within the graph.
- We carefully design a Boundary Token-Aware Prompt for each relation, which further enhances the interaction between boundary tokens and relation semantics.
- Experimental results on two datasets *with* discontinuous entities demonstrate that our method effectively extracts relation triplets, significantly outperforming previous approaches that do not address discontinuity (with an average Accuracy of **17.54** and F1 of **13.45**). Additionally, on two datasets *without* discontinuous entities, our method still outperforms state-of-the-art methods.

Related Work

Zero-Shot Relation Triplet Extraction

The existing ZSRTE methods can be broadly classified into two categories:

Discriminative Methods: These methods assume that all entities are continuous, and focus on identifying entity positions and extracting relation triplets. Lv et al. (2023) employ Soft Prompts for entity and relation extraction, while Lan et al. (2024) treat relations directly as prompts to determine the boundaries of head and tail entities. In contrast, Gong and Eldardiry (2024) improve the comprehension of unseen relations by generating synthetic data for these relations.

Generative Methods: These use Seq2Seq models to generate relation triplets directly. Chia et al. (2022) propose a generative model to generate synthetic data for the unseen relations. Xu et al. (2024) apply reinforcement learning to improve the quality of synthetic data, and Kim et al. (2023) use simple relation templates to extract triplets. Recently, Li et al. (2024) fine-tuned LLMs with tabular prompts, achieving strong results in few- and zero-shot RTE task.

Discontinuous Entity Recognition

Research on discontinuous entities has primarily focused on the NER domain. To overcome the limitations of the traditional BIO tagging scheme, which cannot recognize discontinuous entities, Tang et al. (2013); Xu et al. (2015); Metke-Jimenez and Karimi (2016); Dai, Karimi, and Paris (2017); Tang et al. (2018) propose BIO tag set extensions. Subsequently, methods such as transition-based approaches (Dai et al. 2020) and graph-based methods (Wang et al. 2021a) emerge, specifically design to recognize these entities.

While NER methods primarily identify entities without considering their associations or the complex interactions of relations. *This makes NER methods unsuitable for direct use or simple adaptation in ZSRTE.* Entities, as components of triplets, may also be discontinuous. In the ZSRTE domain, discriminative methods (Lv et al. 2023; Lan et al. 2024; Gong and Eldardiry 2024) typically assume entity continuity, while generative methods (Chia et al. 2022; Kim et al. 2023; Xu et al. 2024; Li et al. 2024) have not delved into entity discontinuity, as detailed in the Introduction, this limitation will also be shown in our experimental comparisons.

Methodology

Task Formulation

Given a dataset $D = (\{S_i, T_i\}_{i=1}^{|D|}, R)$, in which S_i represents the i-th input sentence, T_i denotes its output triplet set, and R is the set of predefined relations, RTE is to extract relation triplets $T_i = \{(e_j^{\text{head}}, e_j^{\text{tail}}, r_j)\}_{j=1}^{|T_i|}$ from S_i, where $r_j \in R$. For a discontinuous entity $e = \{(a_k, b_k)\}_{k=1}^{K}$, it consists of K non-contiguous spans, where a_k and b_k represent the left and right boundaries.

ZSRTE involves learning from a seen dataset D^s and generalizing to an unseen dataset D^u, both derived from the original dataset D. During training, only D^s is used, and the relation label sets R^s and R^u are disjoint.

Encoding of Sentences and Boundary Token-Aware Prompts

To effectively capture the implicit information of each relation label $r \in R$, we design a Boundary Token-Aware Prompt τ_r that incorporates the relation description and learnable Boundary Token Markers. As shown in **Figure 2**, the prompt for a relation label r can be formalized as:

$$\begin{aligned}\tau_r = \{&\text{[H]}, \text{[HEAD]}, \text{[/H]}, \ldots, \\ &\text{[T]}, \text{[TAIL]}, \text{[/T]}, \ldots\},\end{aligned} \tag{1}$$

where [H], [/H], [T], [/T] are learnable *boundary token markers* used to mark the boundary type (the left or right boundary) of each token in a span within a head/tail entity, while [HEAD] and [TAIL] serve as placeholders for the head and tail entities.

During the encoding process, the input sentence $S = \{s_1, s_2, \ldots, s_l\}$, is concatenated with the prompt τ_r, producing a sequence $S' = \{\text{[CLS]}\, S\, \text{[SEP]}\, \tau_r\, \text{[SEP]}\}$, where [CLS] and [SEP] are special tokens. The encoder then generates the token embeddings:

$$H_{S'} = \text{Encoder}(S'). \tag{2}$$

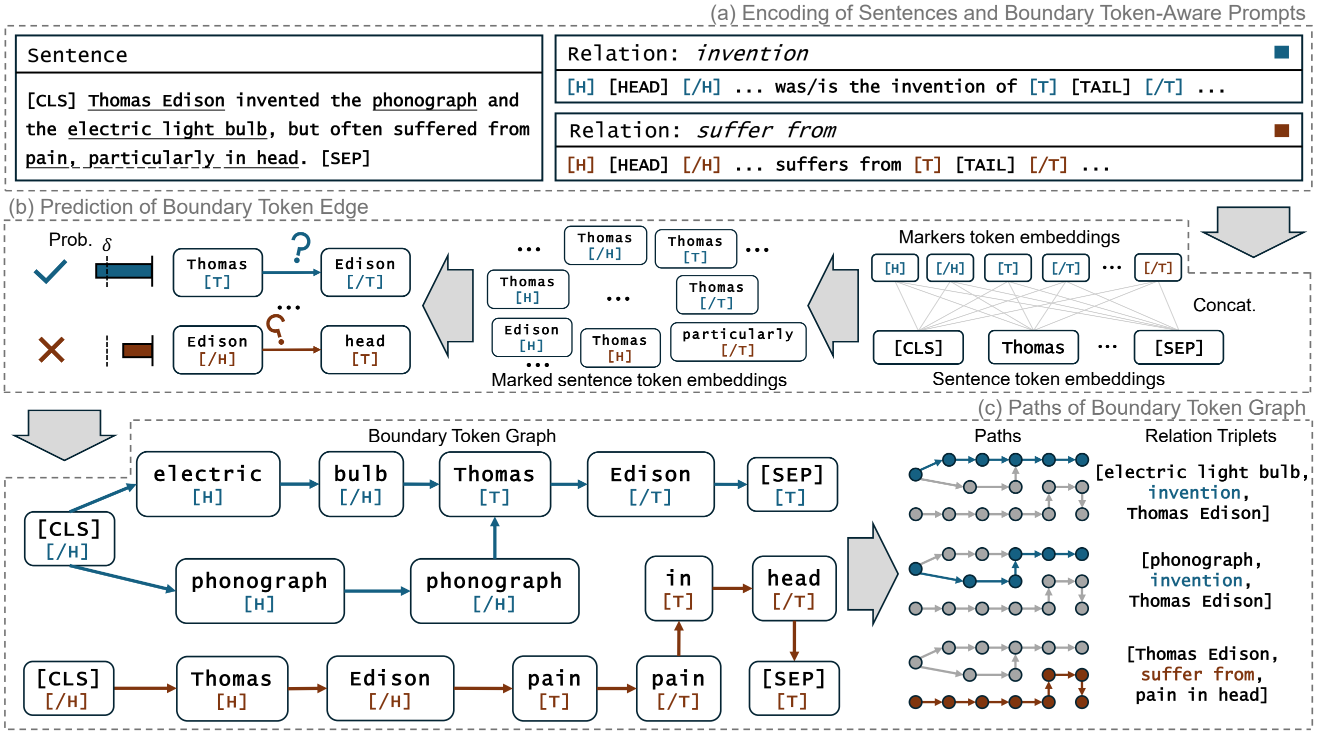


Figure 2: Overview of the proposed BoG framework, consisting of **three main modules**, illustrated using a sentence and two relations 'invention' and 'suffer from'. **First**, we design a `Boundary Token-Aware Prompt` with four special `boundary token markers` ([H], [/H], [T], [/T]) for a relation, and concatenate it with the sentence to obtain token embeddings. **Second**, each token in the sentence is paired with four boundary token markers, and each marked token (e.g., Thomas [H], Thomas [/H]) generates a vertex. Next, edges of different types (Span-Intra Edges and Span-Inter Edges) are predicted between vertex pairs to construct the `Boundary Token Graph`. **Finally**, the ZSRTE task is transformed into finding paths within this graph, where each valid path represents an extracted relation triplet.

For convenience, we denote the set of token embeddings for the input sentence S as $H_S = \{h_{[\text{CLS}]}, h_{s_1}, \ldots, h_{s_l}, h_{[\text{SEP}]}\}$, which includes the embeddings of two special tokens. Additionally, we will later use only the embeddings of the four boundary token markers, $H_M = \{h_{[\text{H}]}, h_{[/\text{H}]}, h_{[\text{T}]}, h_{[/\text{T}]}\}$, which implicitly capture the features of the prompt τ_r.

Prediction of Boundary Token Edge

The goal of Boundary Token Edge Prediction is to determine whether each token serves as the left or right boundary of a span within a head or tail entity, and constructing the Boundary Token Graph. This process involves following *two steps*:

First, we concatenate the boundary marker embeddings H_M with the sentence token embeddings H_S. After the concatenation operation, the set of marked sentence token embeddings V is obtained:

$$V = \{\, t \otimes m \mid t \in H_S,\ m \in H_M \,\}, \tag{3}$$

where $\otimes$ denotes the concatenation operation. In the following, the set V will be treated as the set of vertices. As shown in **Figure 2**, each vertex takes the form of Thomas [H], Thomas [/H], Thomas [T], Thomas [/T], and so on. Furthermore, we divide V into four subsets: $V^{[\text{H}]}$, $V^{[/\text{H}]}$, $V^{[\text{T}]}$, and $V^{[/\text{T}]}$, representing the sets of vertices marked by the four types of boundary token markers, respectively.

Second, we need to predict the directed edges between all vertices, where each edge represents the relationship between boundary tokens. The specific relationships of the edges between different types of marked vertices are shown in **Table 1**. For two vertices v_i and v_j, the prediction probability of a directed edge from v_i to v_j is given by:

$$p_{ij} = \sigma(\text{FFN}(v_i)^\top \text{FFN}(v_j)) \in \mathbb{R}, \tag{4}$$

where FFN denotes a two-layer feedforward network, and σ is the sigmoid activation function. Based on the predicted probability p_{ij} and a predefined edge threshold δ, a directed edge is added from vertex v_i to vertex v_j if $p_{ij} \geq \delta$.

Paths of Boundary Token Graph

After predicting all directed edges, as shown in **Figure 2**, the resulting graph structure is referred to as the Boundary Token Graph. The special tokens [CLS] and [SEP] are incorporated into the graph as the uniform source and sink vertices, respectively. It is important to note that, to minimize unnecessary impacts, we only consider [CLS] marked by [/H] and [SEP] marked by [T]. At this point, each path from the source to the sink in the graph corresponds precisely to a relation triplet. Since each vertex on the path represents a boundary token of a span, the head and tail entities can be identified based on the marker types of these

Type of Edges	Type of Vertices		Relationships
	From	To	
Span-Intra Edges	$V^{[H]} \Longrightarrow$	$V^{[/H]}$	These edges extract spans within head and tail entities, with *an edge* from a span's left boundary token to its right boundary token *indicating the span's boundaries*.
	$V^{[T]} \Longrightarrow$	$V^{[/T]}$	
Span-Inter Edges	$V^{[/H]} \Longrightarrow$	$V^{[H]}$	These edges address cases of discontinuous head or tail entities, with an edge from one span's right boundary token to the next span's left boundary token indicates that *the spans belong to the same discontinuous entity*.
	$V^{[/T]} \Longrightarrow$	$V^{[T]}$	
	$V^{[/H]} \Longrightarrow$	$V^{[T]}$	These edges determine whether *the head and tail entities have the current candidate relation*, with an edge from the right boundary token of the head's last span to the left boundary token of the tail's first span indicating this relation.

Table 1: The specific relationships of the edges between different types of marked vertices are defined. We predict only the probabilities of the five edge types, categorizing them into **Span-Intra Edges** and **Span-Inter Edges**.

boundary tokens, enabling the extraction of the final triplet. In path searching, we use Breadth-First Search or Depth-First Search algorithms to search paths.

Subsequently, all valid paths in the Boundary Token Graph are enumerated to extract triplets corresponding to candidate relations. For *multi-triplet* scenarios, all paths are retained as the result. For *single-triplet* scenarios, due to the varying lengths of the paths, the path with the highest average probability is selected as the final result:

$$E^{*} = \arg\max_{E} \frac{\sum_{\langle v_i, v_j \rangle \in E} p_{ij}}{|E|}, \tag{5}$$

where E represents the set of edges constituting the path, $|E|$ denotes the number of edges in the path, $\langle v_i, v_j \rangle$ represents a directed edge from vertex v_i to v_j, and p_{ij} is the predicted probability of the directed edge $\langle v_i, v_j \rangle$ in Eq.(4).

Training Objective

During the training phase, the objective is to optimize the model to maximize the probability of predicting positive edges while minimizing the probability of predicting negative edges. For the predicted edge probability $p \in P$ between vertices, the Binary Cross-Entropy (BCE) loss is employed:

$$\begin{aligned} \mathcal{L}(P) = -\sum_{v_i \in V} \sum_{v_j \in V} & (y_{ij} \log(p_{ij}) + \\ & (1 - y_{ij}) \log(1 - p_{ij})), \end{aligned} \tag{6}$$

where V denotes the set of vertices in the Boundary Token Graph, p_{ij} represents the predicted probability of a positive edge from vertex v_i to vertex v_j, and $y_{ij} = 1$ indicates the presence of a positive edge from v_i to v_j, while $y_{ij} = 0$ indicates its absence. Since there are two types of edges, a balance hyperparameter α is introduced:

$$\mathcal{L} = \alpha \sum_{P \in P^{\text{intra}}} \mathcal{L}(P) + (1 - \alpha) \sum_{P \in P^{\text{inter}}} \mathcal{L}(P), \tag{7}$$

where P^{intra} represents the predicted probabilities for all Span-Intra Edges, and P^{inter} represents the predicted probabilities for all Span-Inter Edges.

Dataset	# Triplets	# Disc.T	Avg.L
Wiki-ZSL	127,580	0	24.85
FewRel	55,894	0	24.95
Disc-Wiki-ZSL	127,580	19,445 (15.2%)	26.19
Disc-FewRel	55,894	8,638 (15.5%)	26.45

Table 2: The statistics for four ZSRTE datasets. *Disc.T*: triplets involving discontinuous entities; *Avg.L*: average sentence length. Numbers in parentheses indicate the percentage of *Disc.T* relative to the total number of triplets.

Experiments

Datasets

We conducted experiments on three categories of datasets, totaling *five datasets*. The statistics for four ZSRTE datasets are shown in Table 2.

ZSRTE Datasets *without* Discontinuous Entities. First, we evaluate the performance of our method in extracting relation triplets under the zero-shot setting using two standard ZSRTE datasets, `Wiki-ZSL` (Chen and Li 2021) and `FewRel` (Han et al. 2018). Consistent with prior work (Chia et al. 2022), for the test set, the number of unseen relations is set to $m \in \{5, 10, 15\}$, while for the validation set, it is fixed at $m = 5$. To reduce the effects of experimental noise, the data is partitioned using five random seeds, producing in five folds, and the average performance is reported.

ZSRTE Datasets *involving* Discontinuous Entities. Considering the current scarcity of ZSRTE datasets involving discontinuous entities, we manually processed the Wiki-ZSL and FewRel datasets to evaluate the effectiveness of our method in extracting relation triplets involving discontinuous entities, naming the resulting datasets `Disc-Wiki-ZSL` and `Disc-FewRel`. Specifically, we first leverage an entity's context for preliminary identification, then analyze its internal word relationships to introduce supplementary information, achieving the desired splitting.

Discontinuous NER Datasets. We evaluate the effective-

Unseen Relations	**Methods**	**Wiki-ZSL**				**FewRel**			
		Single Triplet	**Multi Triplets**			**Single Triplet**	**Multi Triplets**		
		Acc.	$P.$	$R.$	F_1	*Acc.*	$P.$	$R.$	F_1
m=5	TableSequence (Wang and Lu 2020)	14.47	43.68	3.51	6.29	11.82	15.23	1.91	3.40
	RelationPrompt (Chia et al. 2022)	16.64	29.11	31.00	30.01	22.27	20.80	24.32	22.34
	ZETT (Kim et al. 2023)	21.49	35.89	28.38	31.74	30.71	38.14	30.58	33.71
	DSP (Lv et al. 2023)	-	42.70	43.40	43.00	-	40.10	27.00	32.30
	RSED (Lan et al. 2024)	18.40	38.14	36.84	37.48	22.67	43.91	34.97	38.93
	ZS-SKA (Gong and Eldardiry 2024)	**44.00**	66.70	27.24	38.68	32.86	57.50	26.24	36.04
	TAG (Xu et al. 2024)	23.12	39.36	37.51	38.24	28.94	37.56	40.24	38.81
	Re-Cent (Li et al. 2025)	43.32	53.90	58.55	**55.66**	**46.18**	46.88	44.56	**44.80**
	BoG (ours)	38.00	64.03	44.03	50.16	42.87	49.13	41.62	44.08
m=10	TableSequence (Wang and Lu 2020)	9.61	45.31	3.57	6.40	12.54	28.93	3.60	6.37
	RelationPrompt (Chia et al. 2022)	16.48	30.20	32.31	31.19	23.18	21.59	28.68	24.61
	ZETT (Kim et al. 2023)	17.16	24.49	26.99	24.87	27.79	30.65	32.44	31.28
	DSP (Lv et al. 2023)	-	26.30	48.00	34.00	-	35.90	27.10	30.90
	RSED (Lan et al. 2024)	22.30	27.09	39.09	32.00	24.91	30.89	29.90	30.39
	ZS-SKA (Gong and Eldardiry 2024)	26.40	45.38	29.27	35.30	34.03	60.48	23.22	33.28
	TAG (Xu et al. 2024)	17.24	31.37	32.53	31.88	28.16	31.04	33.49	32.18
	Re-Cent (Li et al. 2025)	30.30	42.22	50.56	45.95	36.53	39.87	39.10	39.05
	BoG (ours)	**35.23**	55.39	48.65	**50.43**	**45.09**	43.45	44.20	**43.47**
m=15	TableSequence (Wang and Lu 2020)	9.20	44.43	3.53	6.39	11.65	19.03	1.99	3.48
	RelationPrompt (Chia et al. 2022)	16.16	26.19	32.12	28.85	18.97	17.73	23.20	20.08
	ZETT (Kim et al. 2023)	12.78	19.45	23.31	21.21	26.17	22.50	27.09	24.39
	DSP (Lv et al. 2023)	-	27.70	32.40	29.90	-	27.90	25.40	26.60
	RSED (Lan et al. 2024)	21.64	25.37	33.80	28.98	25.14	27.00	23.55	25.16
	ZS-SKA (Gong and Eldardiry 2024)	20.26	31.23	27.20	29.19	23.86	37.29	19.13	25.29
	TAG (Xu et al. 2024)	16.41	26.52	31.34	29.18	22.53	25.35	25.88	25.59
	Re-Cent (Li et al. 2025)	27.06	35.79	45.19	39.75	31.27	30.53	32.53	31.07
	BoG (ours)	**32.89**	43.66	38.36	**40.63**	**43.47**	38.80	38.25	**38.24**

Table 3: Comparison results of the proposed BoG on Wiki-ZSL and FewRel datasets *without* Discontinuous Entities. **Bold** marks the highest score. All baseline results are sourced from the original papers.

ness of our method in handling discontinuous entities in NER tasks using `ShARe 14` (Mowery et al. 2014) dataset.

Experimental Settings and Metrics

In experiments, we employ BERT-base (Devlin et al. 2019) as the backbone model. The loss weight α is fixed at 0.5, while the thresholds for Span-Intra Edges δ_{intra} and Span-Inter Edges δ_{inter} are set to 0.4 and 0.3.

For the evaluation metrics, we follow previous work. For `Single Triplet`, where each sentence contains single triplet, Accuracy (*Acc.*) is used. We evaluate `Multi Triplets` using micro Precision ($P.$), Recall ($R.$), and F1-Score (F_1), where each sentence contains multiple triplets, and apply the same metrics to `Discontinuous NER`.

Baselines

We compare our proposed model with the following strong ZSRTE baselines. Among the **discriminative methods**: DSP (Lv et al. 2023), RSED (Lan et al. 2024), ZS-SKA (Gong and Eldardiry 2024) and Re-Cent (Li et al. 2025). Among the **generative methods**: TableSequence (Wang and Lu 2020), RelationPrompt (Chia et al. 2022), TAG (Xu et al. 2024) and ZETT (Kim et al. 2023). We also compare with the LLMs-based methods, including MICRE (Li et al. 2024) and ChatIE (Wei et al. 2023). For discontinuous NER methods, we compare with Comb (Wang and Lu 2019), Trans$_E$ (Dai et al. 2020), BART-based (Yan et al. 2021), MAC (Wang et al. 2021a), Intra-De (Zhang et al. 2022), W^{2}NER (Li et al. 2022) and RerankNER (Xia et al. 2023).

Results and Analysis

Results of ZSRTE *without* Discontinuous Entities

As shown in **Table 3**, our method BoG achieves significant improvements over previous methods on both public datasets. Specifically, in the evaluation of `Single Triplet`, the method achieves the overall highest Accuracy on the Wiki-ZSL and FewRel datasets. In the m=15 setting, it outperforms the second-highest scores by **5.83** and **12.20** points, respectively. For `Multi Triplets`, the method similarly achieves the highest F1 scores on both datasets, with notable improvements over the second-best results. These results demonstrate that the proposed method is more effective than previous methods in the ZSRTE task.

Results of ZSRTE *involving* Discontinuous Entities

To evaluate the model's ability to extract relation triplets involving discontinuous entities, we conducted experiments

Unseen Relations	Methods	Disc-Wiki-ZSL					Disc-FewRel				
		Only Disc	Single Triplet	Multi Triplets			Only Disc	Single Triplet	Multi Triplets		
		Acc.	*Acc.*	*P.*	*R.*	F_1	*Acc.*	*Acc.*	*P.*	*R.*	F_1
m=5	RelationPrompt (Chia et al. 2022)	20.00	17.48	36.46	35.38	35.78	16.09	21.58	21.03	27.07	23.64
	ZETT (Kim et al. 2023)	8.12	15.85	35.53	20.52	25.33	10.54	22.21	30.15	17.95	22.37
	TAG (Xu et al. 2024)	26.11	19.91	38.49	29.52	33.28	27.09	24.19	28.39	32.21	30.09
	BoG (ours)	**44.05**	**37.21**	60.64	42.37	**47.96**	**38.37**	**42.27**	46.61	37.56	**40.39**
m=10	RelationPrompt (Chia et al. 2022)	17.24	14.16	31.16	28.56	29.77	15.56	19.59	19.94	24.29	21.83
	ZETT (Kim et al. 2023)	5.15	11.95	25.22	18.08	20.23	7.45	18.96	23.31	17.18	19.58
	TAG (Xu et al. 2024)	20.14	16.03	32.25	26.62	29.09	26.61	24.42	25.73	28.50	27.01
	BoG (ours)	**41.33**	**34.51**	52.81	45.98	**47.51**	**41.23**	**44.67**	40.74	42.93	**41.00**
m=15	RelationPrompt (Chia et al. 2022)	13.00	11.42	23.84	21.03	22.33	15.97	18.46	18.17	21.25	19.58
	ZETT (Kim et al. 2023)	3.85	8.63	18.96	11.51	14.17	7.17	18.15	18.92	15.86	17.06
	TAG (Xu et al. 2024)	15.00	12.38	25.04	21.12	22.90	23.30	20.03	22.29	23.86	23.04
	BoG (ours)	**36.62**	**30.92**	40.37	34.67	**36.70**	**41.20**	**43.26**	34.75	38.46	**35.71**

Table 4: Comparison results of the proposed BoG on Disc-Wiki-ZSL and Disc-FewRel datasets *with* Discontinuous Entities. **Bold** marks the highest score. All baseline results are our reproductions, using the implementation settings provided in the original papers.

on the datasets Disc-Wiki-ZSL and Disc-FewRel. Approximately 15% of the triplets in Disc-Wiki-ZSL and Disc-FewRel involve discontinuous entities.

Moreover, to more intuitively demonstrate the performance differences of various methods in handling discontinuous entities, we constructed corresponding subsets from the test set of these datasets, where each sentence contains at least one discontinuous entity (i.e., **Only Disc** part in Table 4), and report the results under the Single Triplet setting for these two subsets. Given the limitations of current discriminative methods in extracting relation triplets involving discontinuous entities, we compare our model with three generative methods: RelationPrompt, ZETT, and TAG.

As shown in **Table 4**, compared to baselines, our method extracts triplets involving discontinuous entities more effectively. In the Multi Triplets m=15 setting, we achieve improvements of **13.8** and **12.67** over the second-highest scores on the Disc-Wiki-ZSL and Disc-FewRel datasets.

Overall, the evaluation results on all datasets show that the performance of each baseline declines more significantly when extracting triplets involving discontinuous entities, revealing their limitations in this regard.

Results of Discontinuous Named Entity Recognition

For the discontinuous NER task, we adapt our method primarily by modifying the process of extracting head and tail entities to focus on extracting a single entity, with corresponding adjustments made to the remaining components. Additionally, Boundary Token-Aware Prompts containing relevant descriptions are constructed based on entity types.

As shown in **Table 5**, our method demonstrates *competitive performance* in the discontinuous NER task. While it lags behind the strongest baselines, this is because *our method is not specifically designed for the discontinuous NER task, unlike the baselines*. Moreover, as our approach is primarily tailored for the zero-shot setting, some performance is sacrificed under fully-supervised setting. Nevertheless, the results further validate the effectiveness of our method in handling discontinuous entities.

Methods	ShARe 14		
	P.	*R.*	F_1
Comb (Wang and Lu 2019)	79.10	70.70	74.70
Trans$_E$ (Dai et al. 2020)	78.10	81.20	79.60
BART-based (Yan et al. 2021)	77.20	83.75	80.34
MAC (Wang et al. 2021a)	78.20	84.70	81.30
Intra-De (Zhang et al. 2022)	77.88	83.77	80.72
W^{2}NER (Li et al. 2022)	79.88	83.71	**81.75**
RerankNER (Xia et al. 2023)	78.68	83.63	81.01
BoG (ours)	81.90	80.67	81.28

Table 5: Discontinuous NER experiment of BoG. All baseline results are sourced from the original papers.

Comparison with LLMs-based Methods

Given the remarkable capabilities of large language models (LLMs) in zero-shot learning, we compare our proposed method with eight LLM-based baselines. **Table 6** summarizes the performance of eight baselines along with parameter sizes of their respective backbone models. The results demonstrate that our method achieves superior performance while utilizing significantly fewer parameters.

Ablation Studies

As shown in **Table 7**, we report the impact of each component under the Multi Triplets m=15 setting on FewRel and Disc-FewRel datasets:

- *Boundary Token-Aware Prompts* can express the implicit information of relations. Replacing them with relation labels resulted in a significant performance drop, indicating

Methods	**Backbone Params**	**Wiki-ZSL**			**Disc-Wiki-ZSL**		
		m=5	m=10	m=15	m=5	m=10	m=15
RelationPrompt$_{BART\&GPT-2}$ (Chia et al. 2022)†	264M	16.64	16.48	16.16	17.48	14.16	11.42
ZETT$_{T5-base}$ (Kim et al. 2023)‡	220M	21.49	17.27	12.78	15.85	11.95	8.63
TAG$_{BART\&GPT-2}$ (Xu et al. 2024)†	264M	23.12	17.24	16.41	19.91	16.03	12.38
MICRE$_{T5-3B}$ (Li et al. 2024)‡	3B	25.20	23.65	21.80	-	-	-
MICRE$_{LLaMA}$ (Li et al. 2024)‡	7B	27.74	24.64	22.23	-	-	-
ChatIE$_{GPT-3.5-turbo}$ (Wei et al. 2023)	-	17.19	14.44	11.01	10.02	9.68	9.36
ChatIE$_{GPT-4o}$ (Wei et al. 2023)	-	31.13	20.79	15.74	17.63	13.50	13.00
ChatIE$_{DeepSeek-R1}$ (Wei et al. 2023)	671B	33.42	27.73	22.81	19.60	19.27	15.78
BoG$_{BERT-base}$ (ours)	110M	**38.00**	**35.23**	**32.89**	**37.21**	**34.51**	**30.92**

Table 6: Accuracy comparison results with *LLMs-based methods* on Wiki-ZSL and Disc-Wiki-ZSL datasets under the Single Triplet setting. We report the backbone models used in their baselines along with the corresponding number of parameters (the parameter size of GPT series used by ChatIE (Wei et al. 2023) is not publicly disclosed). † and ‡ respectively mark the results on Wiki-ZSL reported by Xu et al. (2024) and Li et al. (2024), while the remaining results are our reproduction, using the implementation settings provided in the original paper.

Methods	**FewRel**	**Disc-FewRel**
BoG	**38.24**	**35.71**
w/o. Boundary Token-Aware Prompts	25.44	23.80
w/o. Boundary Token Markers	21.40	19.46
w/o. Span-Intra Edges	24.20	22.08
w/o. Span-Inter Edges	35.57	27.54

Table 7: Ablation studies of BoG.

that leveraging these implicit information can provide a more comprehensive understanding of unseen relations.

- *Boundary Token Markers* are utilized to mark the boundary tokens of a span within head and tail entities in a sentence. Removing this component hinders the model's ability to distinguish these boundaries, leading to a decline in performance.
- *Span-Intra Edges* identifies spans that belong to entities. Ignoring the predicted probabilities of this component and indiscriminately connecting all boundary tokens causes the model to fail in correctly identifying head and tail entities. This leads to the generation of numerous incorrect triplets and a significant decline in performance.
- *Span-Inter Edges* represents the connections between different spans, removing these edges means that the relation between the head and tail entities cannot be correctly determined. And directly combining head-tail entity pairs to form triplets generates a substantial number of incorrect triplets, leading to a performance decline. In addition, this component is essential for extracting discontinuous entities, and its removal leads to further performance degradation.

Analysis on Computational Efficiency

To analyze the computational efficiency of our method, we compare it with several existing approaches under the same experimental setting. As shown in **Figure 3**, RelationPrompt and TAG require additional training on synthetic data, which affects their efficiency to some extent. ZETT's use of vocabulary constraints and beam search also impacts decoding efficiency, leading to the increase in computational cost. Our method BoG maintains a significant advantage in computational efficiency.

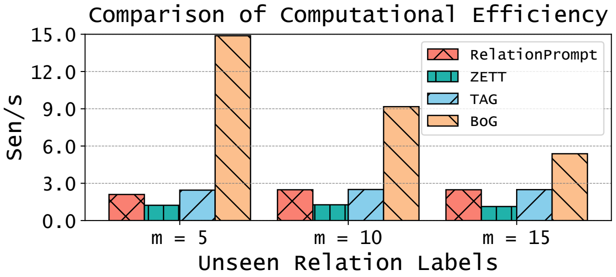


Figure 3: Comparison results of model *computational efficiency* on Disc-FewRel dataset. *Sen/s* refers to the number of sentences can be processed per second.

Conclusion

In this paper, we present the first systematic study of the ZSRTE task involving discontinuous entities and introduce the novel BoG framework, which leverages the Boundary Token Graph structure. By predicting and connecting edges between boundary tokens, our method transforms the relation triplet extraction task into a process of finding paths within a graph, effectively capturing the complex relationships between entities. The introduction of the Boundary Token-Aware Prompt further strengthens the interaction between boundary tokens and relation semantics, thereby improving model performance. Extensive experiments on multiple datasets, including those with and without discontinuous entities, demonstrate that our approach consistently outperforms previous methods, setting new state-of-the-art results in ZSRTE.

Acknowledgements

The authors sincerely thank the reviewers for their valuable comments, which improved the paper. The work is supported by the National Natural Science Foundation of China (62276057).

References

Chen, C.-Y.; and Li, C.-T. 2021. ZS-BERT: Towards Zero-Shot Relation Extraction with Attribute Representation Learning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 3470–3479.

Chia, Y. K.; Bing, L.; Poria, S.; and Si, L. 2022. RelationPrompt: Leveraging Prompts to Generate Synthetic Data for Zero-Shot Relation Triplet Extraction. In *Findings of the Association for Computational Linguistics (ACL)*, 45–57.

Corro, C. F. 2024. A Fast and Sound Tagging Method for Discontinuous Named-Entity Recognition. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 19506–19518.

Dai, X.; Karimi, S.; Hachey, B.; and Paris, C. 2020. An Effective Transition-based Model for Discontinuous NER. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 5860–5870.

Dai, X.; Karimi, S.; and Paris, C. 2017. Medication and Adverse Event Extraction from Noisy Text. In *Proceedings of the Australasian Language Technology Association Workshop*, 79–87.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 4171–4186.

Gong, J.; and Eldardiry, H. 2024. Prompt-based Zero-shot Relation Extraction with Semantic Knowledge Augmentation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (COLING)*, 13143–13156.

Han, X.; Zhu, H.; Yu, P.; Wang, Z.; Yao, Y.; Liu, Z.; and Sun, M. 2018. FewRel: A Large-Scale Supervised Few-Shot Relation Classification Dataset with State-of-the-Art Evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 4803–4809.

Kim, B.; Iso, H.; Bhutani, N.; Hruschka, E.; Nakashole, N.; and Mitchell, T. 2023. Zero-shot Triplet Extraction by Template Infilling. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (IJCNLP-AACL)*, 272–284.

Lan, Y.; Li, D.; Zhang, Y.; Zhao, H.; and Zhao, G. 2024. RSED: Zero-Shot Relation Triplet Extraction via Relation Selection and Entity Boundary Detection. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 11256–11260.

Li, G.; Wang, P.; Liu, J.; Guo, Y.; Ji, K.; Shang, Z.; and Xu, Z. 2024. Meta In-Context Learning Makes Large Language Models Better Zero and Few-Shot Relation Extractors. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI)*, 6350–6358.

Li, J.; Fei, H.; Liu, J.; Wu, S.; Zhang, M.; Teng, C.; Ji, D.; and Li, F. 2022. Unified named entity recognition as word-word relation classification. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 36, 10965–10973.

Li, Z.; Zhang, F.; Lyu, K.; Cheng, J.; and Peng, T. 2025. Re-Cent: A Relation-Centric Framework for Joint Zero-Shot Relation Triplet Extraction. In *Proceedings of the 31st International Conference on Computational Linguistics (COLING)*, 7344–7354.

Lv, B.; Liu, X.; Dai, S.; Liu, N.; Yang, F.; Luo, P.; and Yu, Y. 2023. DSP: Discriminative Soft Prompts for Zero-Shot Entity and Relation Extraction. In *Findings of the Association for Computational Linguistics (ACL)*, 5491–5505.

Metke-Jimenez, A.; and Karimi, S. 2016. Concept Identification and Normalisation for Adverse Drug Event Discovery in Medical Forums. In *BMDID@ISWC*.

Mowery, D. L.; Velupillai, S.; South, B. R.; Christensen, L.; Martinez, D.; Kelly, L.; Goeuriot, L.; Elhadad, N.; Pradhan, S.; Savova, G.; et al. 2014. Task 2: ShARe/CLEF eHealth evaluation lab 2014. In *Proceedings of CLEF 2014*.

Tang, B.; Cao, H.; Wu, Y.; Jiang, M.; and Xu, H. 2013. Recognizing clinical entities in hospital discharge summaries using Structural Support Vector Machines with word representation features. In *BMC medical informatics and decision making*, volume 13, 1–10.

Tang, B.; Hu, J.; Wang, X.; and Chen, Q. 2018. Recognizing Continuous and Discontinuous Adverse Drug Reaction Mentions from Social Media Using LSTM-CRF. *Wireless Communications and Mobile Computing*, 2018(1): 2379208.

Wang, B.; and Lu, W. 2019. Combining Spans into Entities: A Neural Two-Stage Approach for Recognizing Discontiguous Entities. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 6216–6224.

Wang, J.; and Lu, W. 2020. Two are Better than One: Joint Entity and Relation Extraction with Table-Sequence Encoders. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1706–1721.

Wang, Y.; Yu, B.; Zhu, H.; Liu, T.; Yu, N.; and Sun, L. 2021a. Discontinuous Named Entity Recognition as Maximal Clique Discovery. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, 764–774.

Wei, X.; Cui, X.; Cheng, N.; Wang, X.; Zhang, X.; Huang, S.; Xie, P.; Xu, J.; Chen, Y.; Zhang, M.; et al. 2023. ChatIE: Zero-Shot Information Extraction via Chatting with ChatGPT. *arXiv preprint arXiv:2302.10205*.

Xia, Y.; Zhao, Y.; Wu, W.; and Li, S. 2023. Debiasing Generative Named Entity Recognition by Calibrating Sequence Likelihood. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, 1137–1148.

Xu, J.; Zhang, Y.; Wang, J.; Wu, Y.; Jiang, M.; Soysal, E.; and Xu, H. 2015. UTH-CCB: the participation of the SemEval 2015 challenge–Task 14. In *proceedings of the 9th International Workshop on Semantic Evaluation (SemEval)*, 311–314.

Xu, T.; Yang, H.; Zhao, F.; Wu, Z.; and Dai, X. 2024. A Two-Agent Game for Zero-shot Relation Triplet Extraction. In *Findings of the Association for Computational Linguistics (ACL)*, 7510–7527.

Yan, H.; Gui, T.; Dai, J.; Guo, Q.; Zhang, Z.; and Qiu, X. 2021. A Unified Generative Framework for Various NER Subtasks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, 5808–5822.

Zhang, S.; Shen, Y.; Tan, Z.; Wu, Y.; and Lu, W. 2022. De-Bias for Generative Extraction in Unified NER Task. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, 808–818.

Zheng, S.; Wang, F.; Bao, H.; Hao, Y.; Zhou, P.; and Xu, B. 2017. Joint Extraction of Entities and Relations Based on a Novel Tagging Scheme. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, 1227–1236.