LAI: Label Annotation Interaction based Representation Ehancement for End to End Relation Extraction

Anonymous ACL submission

Abstract

While numerous studies on end-to-end relation extraction (E2ERE) have centered on enhancing span representations to improve model performance, challenges remain due to the gaps between subtasks (named entity recognition and relation extraction) and the modeling discrepancies between entities and relations. In this paper, we propose a novel **L**abel **A**nnotation **I**nteraction based representation enhancement method[1]) for E2ERE, which institutes a two-phase semantic interaction to augment representations. Specifically, we firstly feed label annotations that are easy to manually annotate into a language model, and conduct the **first round interaction** between three types of tokens with a partial attention mechanism; Then construct a latent multi-view graph to capture various possible links between label and entity (pair) nodes, facilitating the **second round interaction** between entities and labels. A regimen of rigorous experimentation demonstrates that LAI-Net achieving performance parity with the current SOTA models on ADE/SciERC dataset in terms of NER task (a SOTA performance has been achieved on the ACE05 dataset pecifically), and establishing a new SOTA result (with nearly a **10%** advance on the SciERC dataset for RE specifically) in terms of RE task.

1 Introduction

Endowed with blessing of powerful conversational and generative abilities, large language models (LLMs) like ChatGPT suddenly gained popularity and achieved great success across a spectrum of domains, but recent studies (Han et al., 2023; Li et al., 2023; He et al., 2023; Wang et al., 2023) have unveiled that,in the realm of fundamental NLP tasks and notably in end-to-end relation extraction (E2ERE), LLM still manifests discernible performance discrepancies when juxtaposed with extant SOTA methods.

[1]Our code and models will be publicly available at https://github.com/xxx/xxx.

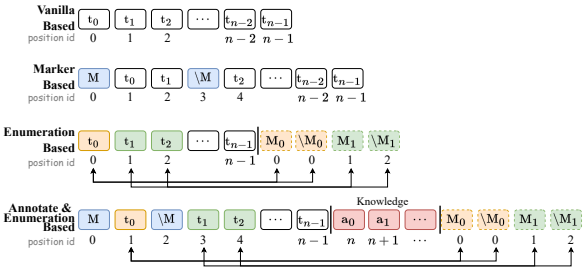

Figure 1: An illustration of different representation enhancement methods. t indicates an individual token from text. The bordered rectangles highlighted in assorted colors signify discrete elements: blue, light yellow (or green), pink respectively indicate entities markers, groups of entity tokens and trailed markers, groups of label annotation tokens. Bidirectionally connected squares sharing the same color refer to elements that have identical position id.

As a core information extraction (IE) task, E2ERE can be split into named entity recognition (NER) subtask for entity identification and relation extraction (RE) subtask for capturing inter-entity relations from plain texts. As postulated by Tang et al. (2022), E2ERE is challenging for its difficulty in capturing affluent correlations between entities and relations. IE research has traditionally converted NER and RE tasks into span-based tasks (Sun et al., 2019; Yang et al., 2021; Tang et al., 2022; Ji et al., 2022; Shang et al., 2022a). Though these methodologies have incrementally advanced model performance from various perspectives, they are still impeded by two pivotal limitations: 1) overdetached of sub-tasks leads to insufficient information exchange between entitiy and relation, and 2) the disparity in modeling strategies between entity and relation result in semantic gaps.

In this paper, we mainly focus on 1) enhancing the semantic interaction during modeling process, and 2) investigating how to obtain a unified and enhanced span representation.

To address the challenges above, prevailing re-

searches mainly focuses on reorganizing the input or intermediate network layers of pre-train language model (PLM), attempting to enhance the semantic information of representation through the integration of specialized symbols or extrinsic prior knowledge. We roughly divided into three types (as shown in Figure.1): **Vanilla based** method is a straightforward approach to acquiring a given span representation by feeding raw text tokens series into pre-trained encoder. **Marker based** method inserts independent entity markers like [M], [\M] amidst text tokens to highlight the presence of entities, aims at attracting more model attention. **Enumerate based** method enumerates all posible entity candidates from plain text and then concatenate them after text tokens, entity tokens share the position ids with candidates as well.

For these methods above, the distinguished LSTM-CRF (Dai et al., 2019) is a typical vanilla based sequence labeling method, and PURE (Zhong and Chen, 2021) is a combination of marker based method and enumeration based method, which adopts marker based method during NER phase and enumerate based method for RE phase, that achieved SOTA performance. PL-Marker (Ye et al., 2022) is a typical enumeration based method that promoted the SOTA further.

In addition to the above three methods, what we develop in this paper can be classified as the fourth class named **annotate & enumeration based** method. It's an novel semantic enhance approach with external knowledge, inspired by external knowledge based aproaches (Sun et al., 2020b; Yang et al., 2021). We argut that a thorough comprehension of label semantics will significantly enhance the IE model abilities, what serves as a premise for our work.

As shown in Figure.1, our principal improvement over preceding methods lies in the insertion of external prior knowledge (i.e., label annotations) into the PLM input sequence, aiming to leverage PLM's internal network layers to enhance semantic interaction between labels and text. This represents the first round semantic interaction in LAI-Net framework.

Unlike the lexicon adapter-based methods (Houlsby et al., 2019; Liu et al., 2021) and lattice-based methods (Zhang and Yang, 2018; Sui et al., 2019; Li et al., 2020), we manually expand the label information and embed it between text tokens and enumerated candidates then feed the series into a pre-trained model. We further enhance the representation by combining word vectors using downstream neural networks.

Formally, the augmented representations derived from the aforementioned four methods can be summarized as:

$$\mathbf{h}^{V}_{\text{span}} = f\left(\mathbf{h}^{s}_{\text{span}}; \mathbf{h}^{e}_{\text{span}}\right)$$
$$\mathbf{h}^{M}_{\text{span}} = f\left(\mathbf{h}_{M}; \mathbf{h}_{\backslash M}\right)$$
$$\mathbf{h}^{E}_{\text{span}} = f\left(\mathbf{h}^{s}_{\text{span}}; \mathbf{h}^{e}_{\text{span}}; \mathbf{h}_{M}; \mathbf{h}_{\backslash M}\right)$$
$$\mathbf{h}^{A\&E}_{\text{span}} = f\left(\mathbf{h}^{s}_{\text{span}}; \mathbf{h}^{e}_{\text{span}}; \mathbf{h}_{M}; \mathbf{h}_{\backslash M}; \mathbf{h}^{s}_{a}; \mathbf{h}^{e}_{a}\right)$$

where $f(\cdot)$ is an tensor operator, equipped to execute series of operations including tensor addition, tensor multiplication, tensor concatenation, etc., or even could be a neural network. And superscripts s, e denote the commencement and termination tokens of a span or annotation, while the subscripts span, $M, \backslash M$ represent the disparate token types.

Based on the first round semantic interaction, we further meticulously crafted with selectively designed downstream network layers. To summarize, our main innovations and improvements in our work lie in the following aspects:

- We explore a novel two-round semantic interaction approach for enhancing span representations, wherein the first round interaction reorganizes the PLM input in what we term an annotation & enumeration-based method, and the second round interaction employs multiple layers of graph convolutional networks built atop Gaussian Graph Generator (GGG) modules to facilitate label semantic fusion.

- We conduct a coarse screening by developing a entity candidate filter, so that achieve the goal of filtering out spans that are clearly not real entities, it also promotes the saving of computing resources.

- Extensive experiment indicates that our model establishes new SOTA performance on several standard benchmarks and surpasses previous models utilizing identical PLM.

2 Related Work

Recently, academic interest in span representation enhancement has surged, providing a substantial impetus to E2ERE. Traditional neural network based methods often ignore non-local and non-sequential context information from input text

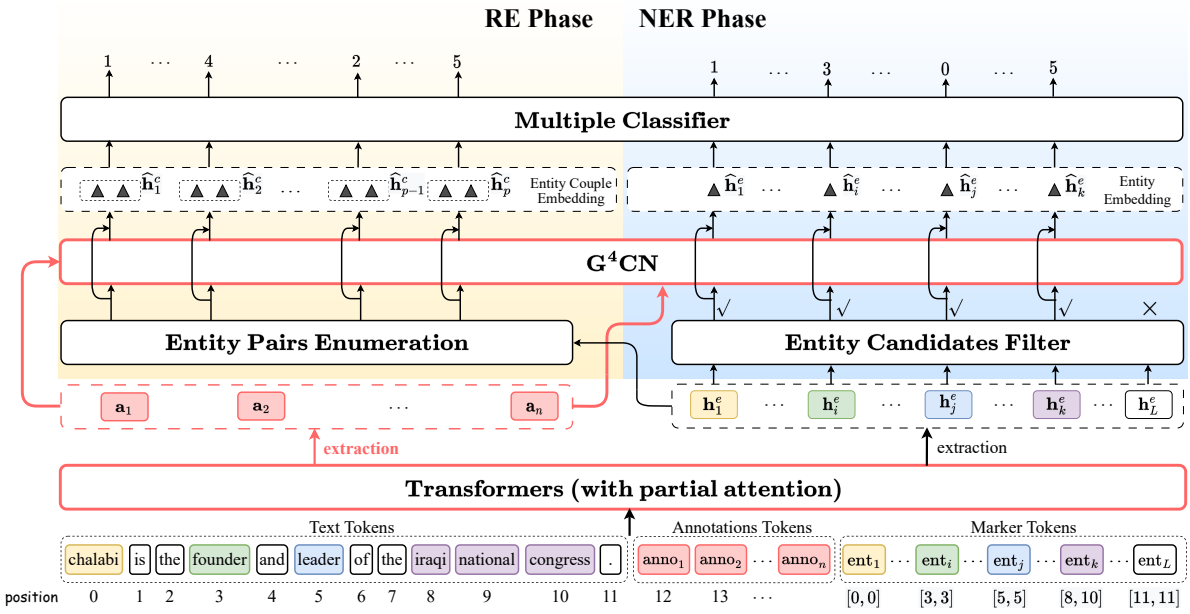

Figure 2: Illustration of our LAI-Net, where annotatation interaction is highlighted by red color. The left and right part of the architecture represent the RE and NER phase respectively. In addition, ▲ indicates the span representation and differed type tokens marked with same color with their corresponding embeddings.

(Qian et al., 2019), what is exactly GCNs (Daigavane et al., 2021; Wu et al., 2023) excel in. GCNs, what our discussion centered on, have been widely used to model the interaction between entities and relations in the text, and has been demonstrated as a typical and effective approach.

GCN-based approaches typically leverage a predefined graph structure, which constructed from plain text, to facilitate information propagation among nodes, thus capturing text's non-linear structure and enhancing NER and RE models' capabilities to capture both global graph structures and representations of nodes and edges.

Mostly GCN based methods (Sun et al., 2019; Guo et al., 2019; Qian et al., 2019; Luo and Zhao, 2020; Sun et al., 2020a; Xue et al., 2021; Shang et al., 2022b) utilizes different approaches to define nodes (sentences, words, tokens, spans, labels, etc.) and edges (syntactic dependency edges, coreference edges, re-occurrence edges, co-occurrence edges, adjacent word edges, etc.). And then perform convolution operations on the graph to facilitate the flow of information between nodes, which enables nodes to efficiently acquire both local and global information. This further refines node representations and downstream network performance. Building on these advances, our work also adopt a GCN based method to design the interaction between entities and relations.

3 Methodology

3.1 Task Definition

Given a sentence formularized as $S = \{w_1, w_2, \cdots, w_m\} = \{t_1, t_2, \cdots, t_n\}$, comprising m words or n tokens ($n \geq m$). The objective of E2ERE is the automated recognization of entity spans and their interrelationships, denotable as $(e_i, r_{i,j}, e_j) \in \mathcal{T}$. Here, e is an entity span, consists of a series of tokens, with an assigned type (e.g., person (PER), organization (ORG)), and $r_{i,j}$ typifies the relationship between e_i and e_j (e.g., ORG-AFF). We define $\mathcal{E}$ and $\mathcal{R}$ as the sets of potential entity and relation types, respectively.

3.2 First Round Semantic Interaction

Before training, we concatenate different type tokens (text tokens, annotation tokens, and marker tokens) sequentially to formulate a unified input sequence (refer to the bottom element of Figure.2). The PLM encoder then conducts the first round semantic interaction, the encoded representation is semantic amalgamation of the three types tokens.

Text Tokens Our approach breaks down the words from raw text into text token sequences as part of the model input.

Annotation Tokens Inspired by Ma et al., 2022 and Yang et al., 2021, we augment semantic information by manually annotating the entity

(or relation) abbreviated label both in NER and RE phase. For example, the abbreviated entity type `GPE` can be annotated as "geography political entity", a fully-semantic unbroken phrase. Correspondingly, the abbreviated relation type `ORG-AFF` can be annotated as "organization affiliation". Each label is manually expanded to enrich semantic content and then tokenized into annotation tokens (highlighted by red rectangle in bottom of Figure.2), which are appended to the text tokens sequentially.

Marker Tokens We enumerate all potential consecutive token sequences (i.e. entity candidates) not exceeding a predefined limitation of length c (with $c \leq n$) within a sentence, labeling each with an entity type. If $c = 2$, as shown in Figure.2, the set of all the possible spans from sentence "`chalabi is the founder and leader of the iraqi national congress.`" can be written as $\mathbf{\Psi} = \{$"chalabi", "chalabi is", "is", "is the", "the founder", "founder", "founder and", "and", "and leader"$\cdots\}$. The i-th span can be written as $\text{span}_i = [\text{span}_i^s, \text{span}_i^e]$, where span_i^s and span_e^e are indicative of start and end position id of entity span respectively. Therefore, entity candidate series can also be written as $\mathbf{\Psi} = \{[0,0],[0,1],[1,1],[1,2],[2,2],[2,3],[3,3],[3,4],[4,4],[4,5]\}$ given position id perspective. Thus, we can easily sumarize the formula for computing the number of candidate for a sentence with m words:$|\mathbf{\Psi}| = m \cdot c + (c - c^2)/2$.

In model input, we define a start marker (M) and an end marker ($\backslash M$), which form a pair of marker tokens, respectively represent the start and end of an entity span and are appended subsequent to annotation tokens. The start and end marker share the same position embedding with corresponding span's start token and end token respectively, while keeping the position id of original text tokens unchanged. From an PLM encoder input perspective, every marker is a token element of tokens series, called marker token. As shown in Figure.2, entity `chalabi` is highlighted by light yellow bordered square, and its corresponding markers noted by a colorful non-bordered square with line frame differred in various entity labels (white means non-entity). In conclusion, the complete input sequence can be represented as follows:

$$\begin{aligned}\widetilde{S} &= \{s_0, s_1, \cdots, s_{|\widetilde{S}|}\} \\ &= \{[\text{CLS}]\} \cup \{t_0, t_1, \cdots, t_{n-1},\} \cup \{[\text{SEP}]\} \\ &\quad \cup \{a_0, a_1, \cdots, a_{N-1}\} \cup \{[\text{SEP}]\} \\ &\quad \cup \{\text{M}_0^s, \text{M}_1^s, \cdots, \text{M}_{|\mathbf{\Psi}|-1}^s\} \cup \{\text{M}_0^e, \text{M}_1^e, \cdots, \text{M}_{|\mathbf{\Psi}|-1}^e\}\end{aligned}$$

where a_i is a single token broken from label annotation, $\text{M}_i^s, \text{M}_i^e$ represent start and end marker token respectively.

Partial Attention Although special tokens (`[CLS]`, `[SEP]`) serve to isolate different types of tokens, there still exists semantic interference among them. The straightforward blend of annotation tokens and marker tokens with text tokens may disrupt semantic consistency of raw text. To mitigate this, we devise a partial attention mechanism, allowing selective semantic influence among the differ types of token. This mechanism can effectively control the information flow (could be regard as a kind of visibility) between different tokens, by adjusting the value of elements of the attention mechanism mask matrix. It suppresses the information interaction among tokens mutual invisible, while enhancing the information interaction among tokens mutual visible. Experimental results show that partial attention effectively improves model performance. See appendix for more detail information about partial attention.

3.3 Second Round Semantic Interaction

Even with partial attention, semantic dissonance persists due to the presence of tripartite token types. To refine semantic integration, we introduce second round semantic interaction, employing a semantic integrator that explicitly model interactions between entity candidates and label annotations. The semantic integrator consists of multiple GCN layers with randomly generated adjacency matrix, treats both entity spans and label annotations as nodes, and establishs connections between nodes through the construction of a graph $\mathcal{G}^s$. So that the interactions between nodes can be explicitly modeled.

A GCN typically necessitates a manually predefined and fixed adjacency matrix to depict the inter-nodes connections. The fixed adjacency matrix fixes the perspective from which the model understands the semantics. However, it's naturally to note that the inter-nodes connection cannot be predetermined accurately when considering our task. Otherwise, the our task would be meaningless. Therefore, we forgo a static adjacency matrix in favor of a multi-view graph, called **G**aussian **G**raph **G**enerator based **G**raph **C**onvolutional **N**etworks, **G^{4}CN** (inspired by He et al., 2015; Xue et al., 2021), and attaches every node with a Gaussian

distribution $\mathcal{N}(\mu, \sigma)$ (μ, σ are generated by a linear layer) to simulate edge weight w_{ij}^e through coumpute the KL divergence w_{ij}^e between two Gaussian distribution of node. This approach captures the potential asymmetrical inter-nodes connections, allows model to assimilate semantic contexts from multiple perspectives, can be formulized as:

$$w_{ij}^e = \mathrm{KL}\left(\mathcal{N}(\mu_i, \sigma_i) \| \mathcal{N}(\mu_j, \sigma_j)\right)$$

$$\widetilde{\mathbf{H}}_{\text{span}} = \frac{1}{2}\left[\mathbf{H}_{\text{span}} + \mathbf{G}^4\mathbf{CN}\left(\mathbf{H}_{\text{span}}, \mathbf{H}_{\text{anno}}^{\text{ent}}\right)\right]$$

where $\mathbf{H}_{\text{span}}, \mathbf{H}_{\text{anno}}^{\text{ent}}$ are matrixs formed by concatenating multiple span or annotation representations.

3.4 Name Entity Recognition

Span and Annotation Representation We extract the contextualized representations $\mathbf{h}$ for individual token s from PLM output, and naturally obtain the involved mathematical formulas for spans and annotatations as:

$$\mathbf{h}_{\text{anno}} = \mathbf{FC}_a\left([\mathbf{h}_a^s; \mathbf{h}_a^e]\right)$$

$$\mathbf{h}_{\text{span}} = \mathbf{FC}_{\text{span}}\left([\mathbf{h}_{\text{span}}^s; \mathbf{h}_{\text{span}}^e; \mathbf{h}_M^s; \mathbf{h}_M^e]\right)$$

where $\mathbf{h}_{\text{anno}} \in \mathbb{R}^d, \mathbf{h}_{\text{span}} \in \mathbb{R}^d$. And $\mathbf{h}_a^s, \mathbf{h}_a^e$ is embedding of first and last token of a certain type of label annotatation, respectively. $\mathbf{h}_t^s, \mathbf{h}_t^e$ is embedding of first token and last token of a entity candidate respectively, and $\mathbf{h}_M^s, \mathbf{h}_M^e$ indicates the embedding of start and end token of marker respectively. Linear layer $\mathbf{FC}$ used to harmonize dimensional space.

Entity Candidates Filter It is indisputable that the prediction of excessive candidate entities significantly consumes a significant amount of computational resources. To conserve computational resources, we devise a binary classifier acts as a entity filter, performing coarse screening for all enumerated entities by discarding non-genuine entities, thus optimizing subsequent predictions.

As for loss function, the primary aim of entity filter is to maximize the likelihood function, what drove us adopt likelihood loss function following Sun et al., 2019:

$$\mathcal{L}_{\text{filter}} = -\frac{1}{|\mathbf{\Psi}|}\sum_{i=1}^{|\mathbf{\Psi}|}\log P(\text{span}_i \in \mathbf{\Psi}_g | \text{span}_i \in \mathbf{\Psi})$$

where $\mathbf{\Psi}_g \subseteq \mathbf{\Psi}$ indicates a set of real entity spans.

In addition to intuitive time consumption optimization, experimental results indicate that entity filter successfully alleviates model weakening engendered by negative samples and enhances overall performance.

Span Classifier We conduct span representations classification through a linear classifier, utilizing cross-entropy loss to direct the learning process. The combined loss function $\mathcal{L}_{\text{ner}} = \mathcal{L}_{\text{filter}} + \mathcal{L}_{\text{span}}$ is optimized during training, with dropout layers for regularization.

$$\mathcal{L}_{\text{span}} = -\frac{1}{|\mathbf{\Psi}_g|}\sum_{\mathbf{\Psi}_g}\log P_{\text{span}}$$

In addition, (Ye et al., 2022) had proved that packing a series of related spans into a training instance can promote the NER model performance, that naturally prompt us to follow the effective measures when reorganize input.

3.5 Relation Extraction

Subject marker In RE phase, we design our Annotate & Enumeration Based method (as demonstrated in Figure.1) to acquire the enhanced representation. Concretely, we adopt the marker based approach (shown in Figure.1), and insert a pair of subject markers (called solid markers in Ye et al., 2022) into left and right of subject entity, and enumerated object candidate spans following on the heels of annotatation tokens to extract relations involving the subject entity.

Entity pairs Representation We match subject and object representations up pairwise to obtain a series of entity pairs, which can be formulized as $\mathbf{h}_{\text{pair}} = [\mathbf{h}_{\text{subj}}; \mathbf{h}_{\text{obj}}]$. And the label semantic confused pair representation formulas is:

$$\widetilde{\mathbf{H}}_{\text{pair}} = \frac{1}{2}\left[\mathbf{H}_{\text{pair}} + \mathbf{G}^{\mathbf{4}}\mathbf{CN}\left(\mathbf{H}_{\text{pair}}, \mathbf{H}_{\text{anno}}^{\text{rel}}\right)\right]$$

where $\mathbf{H}_{\text{pair}}, \mathbf{H}_{\text{anno}}^{\text{rel}}$ are matrixs formed by concatenating entity pair or relation label annotation representations.

4 Experiments

4.1 Experiments Setup

Datasets We utilize three standard corpora: 1) ACE05 spans various domains such as newswire and online forums. It contains seven entity types and six relation types between entities. 2) SciERC(Luan et al., 2018) is a scientific dataset built from AI conference/workshop proceedings across four communities. It includes 7 entity types and 7 relation types. 3) ADE(Gurulingappa et al., 2012) consists of 4,272 sentences and 6,821 relations extracted from medical reports.

Dataset	Models	Encoder	NER			RE			RE+		
			P	R	F1	P	R	F1	P	R	F1
ACE05	Li and Ji(2014)	-	85.20	76.90	80.80	68.90	41.90	52.10	65.40	39.80	49.50
	Miwa and Bansal(2016)	L	82.90	83.90	83.40	-	-	-	57.20	54.00	55.60
	Katiyar and Cardie (2017)		84.00	81.30	82.60	57.90	54.00	55.90	55.50	51.80	53.60
	AntNRE(2019)	G	86.10	82.40	84.20	-	-	-	68.10	52.30	59.10
	DYGIE(2019)		-	-	88.40	-	-	63.20	-	-	-
	DyGIE++(2019)		-	-	88.60	-	-	63.40	-	-	-
	Table-Sequence(2020)		-	-	89.50	-	-	67.60	-	-	64.30
	UniRE(2021)		88.80	88.90	88.80	-	-	-	67.10	61.80	64.30
	SPAN(2020)	Bb	89.32	89.86	89.59	-	-	-	71.22	60.19	65.24
	PURE(2021)		-	-	90.20	-	-	67.70	-	-	64.60
	PL-Marker(2022)		-	-	89.70	-	-	68.80	-	-	66.30
	HIORE(2023)		-	-	89.60	-	-	-	-	-	65.80
	LAI-NET (Our model)		**90.28**	**90.60**	**90.44**	**73.80**	**70.42**	**72.06**	**71.96**	**68.67**	**70.27**
SciERC	DYGIE(2019)	E+L	-	-	65.20	-	-	-	-	-	41.60
	DyGIE++(2019)		-	-	67.50	-	-	-	-	-	48.40
	Spert(2020)		70.87	69.79	70.33	-	-	-	53.40	48.54	50.84
	UniRE(2021)		65.80	71.10	68.40				37.30	36.60	36.90
	PURE(2021)	SciB	-	-	68.20	-	-	50.10	-	-	36.70
	PL-Marker(2022)		-	-	69.90	-	-	52.00	-	-	40.60
	HIORE(2023)		-	-	68.20	-	-	-	-	-	38.30
	LAI-NET (Our model)		70.04	69.89	69.94	**69.21**	**71.71**	**70.41**	**59.84**	**62.01**	**60.88**
ADE	Spert(2020)	Bb	89.02	88.87	88.94	-	-	-	78.09	80.43	79.24
	Table-Sequence(2020)		-	-	89.70				-	-	80.10
	SPAN(2020)		89.88	91.32	90.59	-	-	-	79.56	81.93	80.73
	LAI-NET (Our model)		89.78	91.24	90.49	**80.48**	**83.79**	**82.09**	79.37	**83.28**	**81.25**

Table 1: The main experiment results of overall NER and RE tasks on different datasets. We highlight our results making new SOTA with bold and sub-optimal performance with underline. The Encoder column in the table denotes the base encoder each model used: Bb = BERT-base, SciB = SciBERT (size as BERT-base), E = ELMO, L = LSTM, G = Glove.

Metrics The model with best F1 performance on test set will be selected on a fixed number of epochs. Both micro and macro average metrics are used to evaluate the model performance, former for ACE05/SciERC and latter for ADE. For NER task, an entity prediction is correct if and only if its type and boundaries both match with those of a gold entity. For RE task, a relation prediction is considered correct if its relation type and the boundaries of the two entities match with those in the gold data. We also report the strict relation F1 (denoted RE+), where a relation prediction is considered correct if its relation type as well as the boundaries and types of the two entities all match with those in the gold data. We show detailed experimental settings in Appendix.

4.2 Main Results

4.2.1 Results against horizontal comparison

Table 1 horizontally compares across a spectrum of methods, which focus on comparing outstanding models developed recent years, including several previous SOTA models.

For NER task, given the same PLM encoer, our best model matches or slightly surpasses previous SOTA methods. For RE task, given the same PLM encoder (bert-base), we achieved a performance gain of 2-10% on relation F1 and strict relation F1, consistently outperforming all selected baselines. All these performance comparison results fully demonstrate the advantages of our interactive method.

On ACE05, we can find that LAI-Net achieves an overall lead in terms of NER and RE tasks (**particularly obtains a gain of nearly 4% in the RE task**) compared to all the baselines listed. Both on SciERC and ADE, although our LAI-Net slightly lags behind the SOTA performance in numerical (**nearly 0.4% and 0.1% respectively behind, but still achieves sub-optimal performance**) for NER task, surpasses existing SOTA algorithms (**about 2% and 10% respectively**) in the downstream RE task.

All these improvements demonstrate that 2-rounds semantic interactions indeed further utilizes the predicted entities from the NER stage through the two-rounds semantic interaction, significantly improving the performance of relation recognition without compromising NER performance.

4.2.2 Results against significant hyperparameters

Table 2 delineates the impact of varying significant hyperparameter of GCN layer on the performance of LAI-NET. For the number of GCN layers, we consider a range from 0 to 5, with zero indicating the absence of GCN for semantic interaction and

Task		Number of GCN Layer						Number of Attention Head				
		0	1	2	3	4	5	1	2	3	4	6
ACE05	NER	90.23	89.95	**90.44**	89.92	90.03	89.94	90.16	90.21	90.30	**90.44**	90.24
	RE	68.05	68.38	**72.06**	69.37	69.26	69.53	71.75	72.06	**72.16**	71.84	71.37
	RE+	65.61	65.75	**70.27**	66.93	66.70	66.88	69.54	**70.27**	70.03	69.86	69.21
ADE	NER	**90.49**	90.18	90.23	90.32	90.28	90.17	-	-	-	-	-
	RE	80.99	**82.09**	81.64	80.71	81.25	81.04	81.42	81.79	**82.09**	81.21	80.94
	RE+	80.99	**81.25**	80.81	80.39	80.95	80.63	80.83	81.02	**81.25**	80.88	80.55
SciERC	NER	69.40	**69.94**	69.47	69.17	69.31	69.40	69.42	69.76	69.32	**69.94**	69.23
	RE	69.80	70.06	**70.41**	69.70	70.31	69.36	69.58	69.20	69.99	**70.41**	69.43
	RE+	60.49	60.57	**60.88**	59.91	60.82	60.06	60.24	60.15	60.61	**60.88**	59.56

Table 2: The ablation F1 result comparisons against various configuration setting: (1) if embed entity filter or not, (2) number of GCN layer, (3) number of attention head.

more layers corresponding to increased communication times among nodes within the graph. As for the number of attention heads, we opt for values of 1, 2, 3, 4, and 6. The deliberate exclusion of the value 5 is attributed to the fact that the encoder's hidden dimension is not divisible by 5, a constraint inherent to the multi-head attention mechanism.

The table permits an intuitive observation that: (1) an increase in the GCN layers number does not linearly translate to enhanced performance; an excessive GCN layers number can exert a deleterious effect on the model, with the more optimal layers number identified as either 1 or 2; (2) concerning the number of attention heads, the more optimal solution exhibits some variation across different datasets, yet it is unequivocally clear that neither an excessively high (e.g. 6) nor a disproportionately low (e.g. 1) number of attention heads can fully capitalize on the GCN's capabilities.

4.3 Ablation Study

4.3.1 Performance against two rounds of interaction

We conducted ablation experiment specifically targeting the two-stage semantic interaction, which is considered crucial ablation experiment in our work. Drawing upon the outcomes of the experiment, we can directly evaluate the extent to which our devised dual semantic interaction genuinely augments the model's performance.

What should be noted is that the second stage interaction is built upon the annotated label information (that's what the first stage interaction do), so when the label annotation information is no longer inject, the second stage interaction ceases to exist as well. However, the existence of the second stage interaction does not affect the first stage interaction.

As shown in Table 3, no matter which round of semantic interaction is eliminated, it invariably leads to adverse effects of varying magnitudes on the newly SOTA we have developed, encompassing the precision, recall, and F1-score metrics. Notably, the impact on the RE task for the ACE05 dataset is particularly severe, with the greatest reduction in the F1-score reaching up to 4.67%.

Task		Method	P	R	F1
ACE05	NER	LAI-Net	**90.28**	90.60	**90.44**
		w/o 1st	89.72 (-0.55)	90.68 (+0.08)	90.20 (-0.24)
		w/o 2nd	89.97 (-0.31)	90.50 (-0.10)	90.23 (-0.21)
	RE	LAI-Net	**73.80**	**70.42**	**72.06**
		w/o 1st	70.04 (-3.76)	67.83 (-2.60)	68.91 (-3.15)
		w/o 2nd	69.70 (-4.09)	66.49 (-3.94)	68.05 (-4.01)
	RE+	LAI-Net	**71.96**	**68.67**	**70.27**
		w/o 1st	67.14 (-4.82)	65.02 (-3.66)	66.06 (-4.21)
		w/o 2nd	67.20 (-4.77)	64.10 (-4.58)	65.61 (-4.67)
ADE	NER	LAI-Net	**89.78**	**91.24**	**90.49**
		w/o 1st	88.94 (-0.84)	91.07 (-0.17)	89.99 (-0.51)
		w/o 2nd	-	-	-
	RE	LAI-Net	**79.38**	**83.29**	**81.26**
		w/o 1st	79.08 (-0.30)	82.99 (-0.30)	80.99 (-0.27)
		w/o 2nd	79.01 (-0.37)	83.17 (-0.12)	81.04 (-0.22)
	RE+	LAI-Net	**79.37**	**83.28**	**81.25**
		w/o 1st	78.87 (-0.51)	82.68 (-0.61)	80.73 (-0.52)
		w/o 2nd	78.73 (-0.64)	82.64 (-0.64)	80.63 (-0.62)
SciERC	NER	LAI-Net	**70.04**	**69.89**	**69.94**
		w/o 1st	69.58 (-0.46)	69.12 (-0.77)	69.35 (-0.59)
		w/o 2nd	69.82 (-0.22)	68.98 (-0.91)	69.40 (-0.54)
	RE	LAI-Net	**69.21**	**71.71**	**70.41**
		w/o 1st	69.18 (-0.03)	69.94 (-1.77)	69.56 (-0.85)
		w/o 2nd	69.00 (-0.21)	70.64 (-1.08)	69.80 (-0.61)
	RE+	LAI-Net	**59.84**	**62.01**	**60.88**
		w/o 1st	59.81 (-0.03)	60.47 (-1.54)	60.14 (-0.74)
		w/o 2nd	59.79 (-0.05)	61.22 (-0.80)	60.49 (-0.39)

Table 3: The ablation result comparisons against two rounds interaction.

4.3.2 Ablations against attention mask matrix

In order to evaluate the efficacy of partial attention matrices, we selected three distinct attention masking matrices for ablation study, namely: (1) the full attention matrix, wherein all tokens are mutually visible; (2) the anno-token visible attention matrix, where annotation tokens and text tokens are intervisible; and (3) the anno-token invisible attention matrix, with annotation tokens and text tokens also being intervisible. Irrespective of the

	Task	NER	RE	RE+
ACE05	Invisible	**90.44**	72.06	70.27
	Visible	90.36 (-0.08)	68.01 (-4.05)	65.57 (-4.70)
	Full	88.29 (-2.14)	65.79 (-6.28)	64.28 (-5.99)
ADE	Invisible	**90.49**	82.09	81.25
	Visible	90.09 (-0.40)	81.26 (-0.83)	80.09 (-1.16)
	Full	89.40 (-1.10)	79.99 (-2.10)	78.92 (-2.33)
SciERC	Invisible	**69.94**	70.41	60.88
	Visible	69.13 (-0.81)	69.44 (-0.97)	60.61 (-0.27)
	Full	66.21 (-3.73)	69.55 (-0.85)	60.30 (-0.59)

Table 4: The ablation F1 result comparisons against various types of attention mask matrix.

Entity Filter	ACE05 $_{F1}$	ADE $_{F1}$	SciERC $_{F1}$
w/	**90.44**	**90.49**	**69.94**
w/o	88.70 (-1.74)	89.92 (-0.57)	69.76 (-0.18)

Table 5: The ablation result comparisons against entity filter.

attention masking matrix employed, tokens of the same type remain mutually visible during the computation of attention scores. And a more in-depth explanation has been provided in Appendix.

As Table 4 elucidates, across all three datasets, the anno-token invisible attention matrix exhibits a markedly superior performance compared to the other attention matrice types. The anno-token visible attention matrix comes in second place, with its largest deficit compared to the top-performing technique capping out at 4.7% across the varied tasks encapsulated within the trio of benchmarks. Meanwhile, the fully attention matrix turns in the least impressive showing, lagging behind the peak achieved score on each respective dataset by up to 6.28%, indicative of appreciably inferior capabilities amongst the range of workloads tested.

In attempting to analyze the underlying reasons for this phenomenon, it is hypothesized that the following factors play a critical role: (1) the mutual visibility mechanism between annotation tokens and text tokens establishes a conduit for semantic communication, thereby enhancing the semantic richness of the embeddings for both annotations and text; (2) conversely, the full attention matrix allows for the intermingling of semantic information among annotation tokens, text tokens, and entity candidate tokens, which may inadvertently lead to an over-amplification of semantic input, potentially diluting the primary information or even causing semantic confusion, culminating in a decrement in model performance.

4.3.3 Ablations against entity filter

During the NER phase, considering the exponential surge in candidate entity quantity accompanying increased entity length, the preponderance of negative samples can readily overwhelm the relatively paltry positive examples, which easily impedes the network's capacity to accurately identify genuine entities. To mitigate this, LAI-Net incorporates a deliberately inserted filter prior to the entity classifier to preliminarily sieve out spans that are clearly non-entity. To validate whether said filter genuinely facilitates the NER process, we devised associated ablation experiments. As depicted in Table 5, the presence of the filter effectively improves NER performance, with advantages most conspicuous on the ACE05 dataset (surpassing no-filter models by 1.74% in terms of F1 scores).

5 Conclusion

Faced with the weaknesses of LLMs in fundamental NLP tasks, we propose LAI-Net, a novel interaction-based E2ERE method for representational semantic enhancement via two round interactions. Experiments on several datasets show that our LAI-Net allows high-level semantic information flow and facilitates the E2ERE task.

In summary, the key novelty is the multi-phase semantic interaction framework to effectively inject external knowledge and unify representations for entities and relations. The gains demonstrate this allows better information flow and facilitates the E2ERE task.

6 Ethical Statement

This research strictly adheres to the ethical code outlined by ACL, recognizing the significance of ethical considerations in AI. Our work on relation extraction is conducted with a commitment to responsible use of data and conscious awareness of the potential impacts of our findings. We acknowledge the potential of this technology to influence various AI applications, and we are aware of the dual-use nature of our contributions. To mitigate risks of misuse, we advocate for transparent and regulated use of relation extraction technologies, especially in sensitive domains. Our research is oriented towards positive societal benefits, such as enhancing information accessibility and aiding knowledge discovery, while actively seeking to minimize any adverse consequences.

7 Limitations

Our study, while advancing the capabilities in relation extraction, has certain limitations. Firstly, the performance of our proposed models is highly dependent on the quality of manual annotation of label semantics, especially on fine-grained relational extraction tasks. Although we strive to build reliable and trustworthy annotations during the experiment, it can take a lot of manpower to complete this work in practical applications, potentially limiting their applicability in those areas. Secondly, the token number of label annotatation is also an important factor limiting the performance of the model. Due to the input length limitation of the encoder, the number of words in label annotatation will undoubtedly consume length space, and overlong annotatation may cause greater semantic confusion, which in turn affects the model performance

References

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Zhenjin Dai, Xutao Wang, Pin Ni, Yuming Li, Gangmin Li, and Xuming Bai. 2019. Named entity recognition using bert bilstm crf for chinese electronic health records. In *2019 12th international congress on image and signal processing, biomedical engineering and informatics (cisp-bmei)*, pages 1–5. IEEE.

Ameya Daigavane, Balaraman Ravindran, and Gaurav Aggarwal. 2021. Understanding convolutions on graphs, Understanding the building blocks and design choices of graph neural networks. `https://distill.pub/2021/understanding-gnns/`, Published in 2021-9-2.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Zhijiang Guo, Yan Zhang, and Wei Lu. 2019. Attention guided graph convolutional networks for relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 241–251, Florence, Italy. Association for Computational Linguistics.

Harsha Gurulingappa, Abdul Mateen Rajput, Angus Roberts, Juliane Fluck, Martin Hofmann-Apitius, and Luca Toldo. 2012. Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. *Journal of Biomedical Informatics*, 45(5):885–892. Text Mining and Natural Language Processing in Pharmacogenomics.

Ridong Han, Tao Peng, Chaohao Yang, Benyou Wang, Lu Liu, and Xiang Wan. 2023. Is information extraction solved by chatgpt? an analysis of performance, evaluation criteria, robustness and errors.

Jiabang He, Lei Wang, Yi Hu, Ning Liu, Hui Liu, Xing Xu, and Heng Tao Shen. 2023. Icl-d3ie: In-context learning with diverse demonstrations updating for document information extraction.

Shizhu He, Kang Liu, Guoliang Ji, and Jun Zhao. 2015. Learning to represent knowledge graphs with gaussian embedding. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, page 623–632, New York, NY, USA. Association for Computing Machinery.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Bin Ji, Shasha Li, Hao Xu, Jie Yu, Jun Ma, Huijun Liu, and Jing Yang. 2022. Span-based joint entity and relation extraction augmented with sequence tagging mechanism. *arXiv preprint arXiv:2210.12720*.

Bo Li, Gexiang Fang, Yang Yang, Quansen Wang, Wei Ye, Wen Zhao, and Shikun Zhang. 2023. Evaluating chatgpt's information extraction capabilities: An assessment of performance, explainability, calibration, and faithfulness.

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

Xiaonan Li, Hang Yan, Xipeng Qiu, and Xuanjing Huang. 2020. FLAT: Chinese NER using flat-lattice transformer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6836–6842, Online. Association for Computational Linguistics.

Wei Liu, Xiyan Fu, Yue Zhang, and Wenming Xiao. 2021. Lexicon enhanced Chinese sequence labeling

using BERT adapter. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5847–5858, Online. Association for Computational Linguistics.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3219–3232, Brussels, Belgium. Association for Computational Linguistics.

Ying Luo and Hai Zhao. 2020. Bipartite flat-graph network for nested named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6408–6418, Online. Association for Computational Linguistics.

Jie Ma, Miguel Ballesteros, Srikanth Doss, Rishita Anubhai, Sunil Mallya, Yaser Al-Onaizan, and Dan Roth. 2022. Label semantics for few shot named entity recognition. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1956–1971, Dublin, Ireland. Association for Computational Linguistics.

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

Yujie Qian, Enrico Santus, Zhijing Jin, Jiang Guo, and Regina Barzilay. 2019. GraphIE: A graph-based framework for information extraction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 751–761, Minneapolis, Minnesota. Association for Computational Linguistics.

Yu-Ming Shang, Heyan Huang, Xin Sun, Wei Wei, and Xian-Ling Mao. 2022a. Relational triple extraction: One step is enough. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 4360–4366. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Yu-Ming Shang, Heyan Huang, Xin Sun, Wei Wei, and Xian-Ling Mao. 2022b. Relational triple extraction: One step is enough. *arXiv preprint arXiv:2205.05270*.

Dianbo Sui, Yubo Chen, Kang Liu, Jun Zhao, and Shengping Liu. 2019. Leverage lexical knowledge for Chinese named entity recognition via collaborative graph network. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3830–3840, Hong Kong, China. Association for Computational Linguistics.

Changzhi Sun, Yeyun Gong, Yuanbin Wu, Ming Gong, Daxin Jiang, Man Lan, Shiliang Sun, and Nan Duan. 2019. Joint type inference on entities and relations via graph convolutional networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1361–1370, Florence, Italy. Association for Computational Linguistics.

Kai Sun, Richong Zhang, Yongyi Mao, Samuel Mensah, and Xudong Liu. 2020a. Relation extraction with convolutional network over learnable syntax-transport graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8928–8935.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2020b. Ernie 2.0: A continual pre-training framework for language understanding. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8968–8975.

Wei Tang, Benfeng Xu, Yuyue Zhao, Zhendong Mao, Yifeng Liu, Yong Liao, and Haiyong Xie. 2022. UniRel: Unified representation and interaction for joint relational triple extraction. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7087–7099, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. 2023. Gpt-ner: Named entity recognition via large language models.

Lingfei Wu, Yu Chen, Kai Shen, Xiaojie Guo, Hanning Gao, Shucheng Li, Jian Pei, Bo Long, et al. 2023. Graph neural networks for natural language processing: A survey. *Foundations and Trends® in Machine Learning*, 16(2):119–328.

Fuzhao Xue, Aixin Sun, Hao Zhang, and Eng Siong Chng. 2021. Gdpnet: Refining latent multi-view graph for relation extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14194–14202.

Pan Yang, Xin Cong, Zhenyu Sun, and Xingwu Liu. 2021. Enhanced language representation with label knowledge for span extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4623–4635, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Deming Ye, Yankai Lin, Peng Li, and Maosong Sun. 2022. Packed levitated marker for entity and relation extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4904–4917, Dublin, Ireland. Association for Computational Linguistics.

Yue Zhang and Jie Yang. 2018. Chinese NER using lattice LSTM. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1554–1564, Melbourne, Australia. Association for Computational Linguistics.

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

A Appendices

A.1 Implement Details

Datasets and preprocess We strategically selected three standard corpus (ACE05, SciERC, ADE) in terms of E2ERE task.

1) ACE05[2] is collected from a variety of domains (such as newswire and online forums). It includes 7 entity types, and include 6 relation types between entities. For data processing, we use the same entity and relation types, data splits[3], and preprocessing as Li and Ji, 2014; Miwa and Bansal, 2016 (351 training, 80 development and 80 testing).

2) SciERC (Luan et al., 2018) is a scientific-oriented dataset, which is built from 12 AI conference/workshop proceedings in four AI communities, and includs 7 entity types and 7 relation types.

3) ADE (Gurulingappa et al., 2012) consists of 4272 sentences and 6821 relations extracted from medical reports.

In terms of experiment, for ACE05 and SciERC, we run our model 10 times with different random seeds, and report averaged results of all the runs. And for ADE, we adopt 10 fold cross-validation respectively and run each fold 10 times and report averaged results of all the runs.

PLMs and hardware devices For fair comparison with previous works, we employ bert-base-uncased (Devlin et al., 2019) as the encoders for ACE05 and ADE, and use the in-domain scibert-scivocab-uncased[4] (Beltagy et al., 2019) as the encoder for SciERC, and all the experiments are executed using three GeForce RTX 3090 24GB GPUs.

Optimizer and learning rate settings We use AdamW optimizer during training. We set the learning rate as 4e-4 for both NER and RE task. We had tried to set different learning rate for different layers, and experiment results show that it's useless.

Maximum length settings We respectively set the maximum length of reorganized sentence C as 150, 100, 150 on ACE05, ADE, SciERC. As enumerating possible spans, we set the maximum span length L as 8 for all datasets, and limit the number of entity candidates as 220 for every train/eval sample.

Batch size and epoch settings In NER phase, we respectively set batch size as 16 per GPU for SciERC, 20 per GPU for ACE05, and 14 per GPU for ADE. And in RE phase, we set the batch size as 40 per GPU for SciERC, 50 per GPU for ACE05, 64 per GPU for ADE. We set the epoch as 80 for all Datasets in NER phase, and 60 for all dataset in RE phase.

Cold start settings for NER It should be noted that, due to the enumeration of a large number of non-entity spans as negative samples in NER pahse, which extremely likely to lead to unable-convergence train, or the phenomenon that the model directly predicts all spans as non-entities. Therefore, we set up a specific cold start process for training. In the first half of training, we use the true labels as the filter result for next entity classification in order to calculate the loss to update the parameters. This guides the model during early learning before exposing it to negatively labeled non-entity spans. In actual training, we set the cold boot epoch number as 15, 40, 40 for ACE05, SciERC, ADE respectively. Moreover, according to unpublished experimental results, experiments without cold start configuration were utterly incapable of acquiring any knowledge whatsoever, with all performance metrics equaling zero throughout the training phase.

Symmetry of relation for RE We formulate the directed relation as $r_{i,j}$, with the subject entity e_i always pointing to object entity e_j. Therefore, a triplet with positive relation canbe written as $e_i \rightarrow r_{i,j} \rightarrow e_j$, and its reverse formula is $e_j \rightarrow r_{j,i} \rightarrow e_i$. $r_{i,j}, r_{j,i}$ may refer to different relation types. In RE phase, we consider the subject be left and the object be right by default. Either $(e_i \rightarrow r_{i,j} \rightarrow e_j)$ or $(e_j \rightarrow r_{j,i} \rightarrow e_i)$ will be predicted if it's a really relaiton. Only when they are both predicted to be true (that is, the probability value is greater than the threshold), the triplet $(e_i, r_{i,j}, e_j)$ and triplet $(e_j, r_{j,i}, e_i)$ will be established.

A.2 Parital attention mask

Obviously, the series of annotation tokens and marker tokens manually attached after the text tokens does not form a coherent and semantically complete sentence. It inevitably affects the semantic construction from text tokens, and impair the representational ability of word vectors during the PLM encoding process.

To address this potential issue, we adopt a spe-

[2] https://catalog.ldc.upenn.edu/LDC2006T06

[3] https://github.com/tticoin/LSTM-ER/tree/master/data/ace2005/split

[4] SciBERT is a BERT model trained on scientific text, whose corpus includes the full text of 1.14 million scientific papers (82% in biomedical and 12% in computer science), and may be more suitable for natural language processing tasks on SciERC dataset

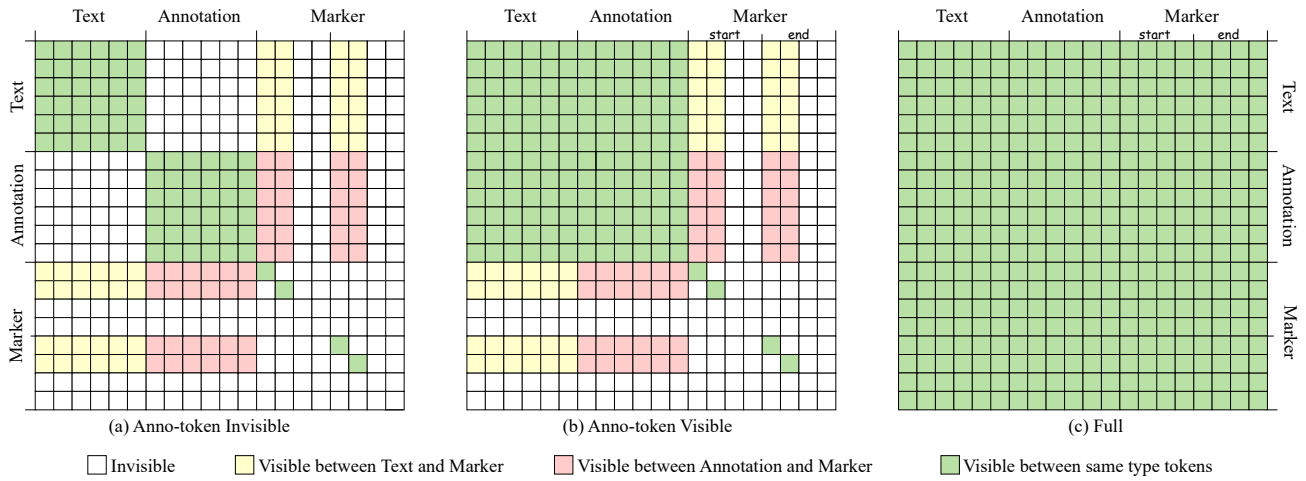

Figure 3: Diagrams of three types attention mask matrix with an assumption that there is two entity candidates.

cialized partial attention mechanism to selectively mitigate or enhance the semantic impact of tokens from differed types. In details, by adjusting the value of elements of the attention mechanism mask matrix, we can control the visibility among three types of tokens.

Partial attention can effectively control the information flow between different tokens. It suppresses the information interaction between text and annotations (i.e. invisible) while enhancing the information interaction between text and candidates (i.e. visible).

We show three different masking matrices in Figure.3, for which we conduct some ablation experiments. In conjunction with Figure.3, we further elucidate the meaning of the discrete elements within mask matrix. Train our gaze upon the first row of Figure.3 (a), which delineates the tokens discernible by the text token. The pale green region signifies it can see corresponding tokens, the white space those it cannot see, and the faint yellow elements the marker tokens within its purview. Notably, the two faint yellow squares on the left denote the start marker tokens, while the two on the right denote the end marker tokens.