SDB-DRE: Learning Structure, Definition and Boundary Makes LLMs Better Document-Level Relation Triplet Extractors

Anonymous ACL submission

Abstract

Recent years, Large Language Models (LLMs) demonstrate superior performance in information extraction tasks. Leveraging these models for Document-Level Relation extraction (DocRE) will benefits from their powerful generative capabilities. However, we observe that LLMs still face challenges in DocRE tasks: Document Structure Parsing Error, Relation Definition Ambiguity, and Entity Boundary Recognition Error. To address these issues, we propose SDB-DRE, an LLM-based DocRE model that does not rely on pre-labeled entities. To tackle the Document Structure Parsing Error, we introduce a novel Structure-Aware QA training approach, enabling LLMs to learn coreference relationships and entity types within the document. To resolve Relation Definition Ambiguity and Entity Boundary Recognition Error, we introduce relation definition learning and mention boundary learning in the second stage of relation extraction training. This improves the internal document representation of the LLM, ensuring the output triples are consistent with the relation definitions and have more accurate entity boundaries. Experimental results show that SDB-DRE outperforms LLM-based methods using multi-stage inference in a single-stage reasoning setup, achieving state-of-the-art performance.

1 Introduction

Document-level Relation Extraction (DocRE) (Yao et al., 2019; Xie et al., 2022) focuses on extracting relationships between entities from the given document. Compared to sentence-level task, DocRE is more complex due to phenomena like coreference and cross-sentence relations, but it more accurately reflects practical applications. Previous research (Tan et al., 2022a) mostly relies on pre-annotated entities for entity-pair relation classification, which does not fully capture the complexities of real-world scenarios. In light of this, some studies(Eberts and Ulges, 2021; Xu and Choi, 2022; Zhang et al., 2023) shift towards more complex settings document-level joint relation extraction, where the model simultaneously solve entity mentions recognition, coreference resolution and relation extraction. However, such approaches overly refine the steps involved in DocRE, leading to accumulated errors that degrade relation extraction performance. Recent advancements (Jiang et al., 2023; Achiam et al., 2023) in the area of Large Language Models (LLMs) make it feasible to build end-to-end triplet extraction models. Leveraging LLMs for DocRE allows for the utilization of extensive pre-trained knowledge and powerful generative capabilities. This motivates us to explore how LLMs can be better applied to DocRE.

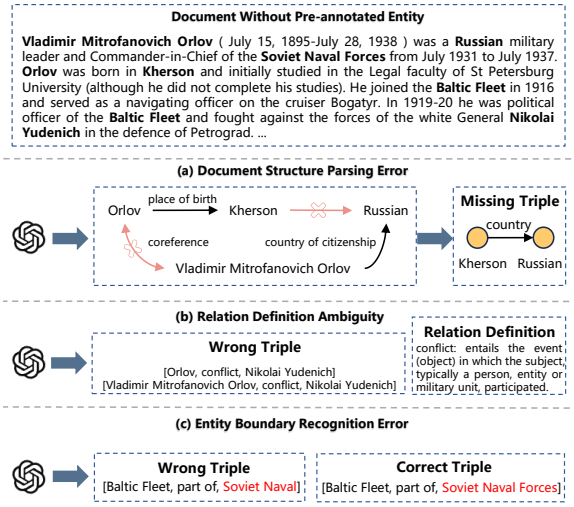

Figure 1: An example of end-to-end DocRE task based on LLMs and three common errors: Document Structure Parsing Error, Relation Definition Ambiguity, and Entity Boundary Recognition Error.

Despite promising results from recent LLM-based methods (Xue et al., 2024), current multi-step inference methods increase computational costs. In addition, these methods have three typical problems: (1) Document Structure Parsing Error (2) Relation Definition Ambiguity and (3) Entity

Boundary Recognition Error. As shown in Figure 1, we select a document example to more intuitively demonstrate the three phenomena produced by LLMs and their underlying causes, when performing end-to-end triplet extraction.

Document Structure Parsing Error. The trained LLMs can correctly predict that *Orlov*'s birthplace is Kherson and that Vladimir Mitrofanovich Orlov's *country of citizenship* is *Russian*. However, due to the lack of document structure parsing ability in LLMs for specific scenarios, it fails to recognize both *Orlov* and*Vladimir Mitrofanovich Orlov* actually refer to the same entity, thus preventing further inference of the *country* relationship between *Kherson* and *Russian*. This limitation stems from the model not being explicitly endowed with document structure parsing capabilities during training.

Relation Definition Ambiguity. From the perspective of document semantics and relation name conflicts, the relation [*Orlov, conflict, Nikolai Yudenich*] appears to be correct. However, the model overlooks the specific definition of *conflict*, which requires the object entity to be an event and the subject entity to participate in it (e.g., *Winston Churchill, conflict, World War II*). There are numerous relation categories in DocRE. Therefore, how to enable the LLM to accurately understand relation definitions in specific scenarios remains a critical challenge.

Entity Boundary Recognition Error. This errors in triplet prediction represent another typical issue. For instance, the LLM's incorrect boundary recognition of the *Soviet Naval Forces* leads to erroneous triplets even when the relations and document semantics are correctly understood by LLM. Improving the accuracy of entity mentions boundary recognition in end-to-end output is crucial for enhancing LLM performance.

To address these issues, we propose a single-stage inference, LLM-based document-level triplet extraction method called SDB-DRE (Structure, Definition and Boundary-Document Level Relation Extraction). Specifically, in the first-stage training, we construct Structure-Aware Question Answer (SAQA) data which includes entity categories and coreference parsing QA pairs to enhance LLM's foundational document structure parsing ability. Building on the second-stage relation extraction training, we introduce two additional learning mechanisms for LLMs: Relation Definition learning and Mention boundary learning. The former introduces relation definition memory during training to mitigate the negative impact on the model performance when the number of relation types becomes too large and the model struggles to comprehensively understand relation definitions. The latter enhances the accuracy of LLMs in identifying the boundaries of head and tail entities when outputting triples. Notably, our method does not rely on pre-given entity annotations and can directly perform end-to-end triple extraction during inference, offering greater generalizability and potential for practical applications. Our contributions can be summarized as follows:

- We identify three significant issues in existing LLM-based end-to-end DocRE methods: (1) Document Structure Parsing Error (2) Relation Definition Ambiguity and (3) Entity Boundary Recognition Error.
- We propose a single-stage Inference DocRE model based on LLMs, called SDB-DRE, which includes the following key components: (1) The SAQA training enhances the LLM's ability to parser document structure (2) Static relation definition memory learning enables the LLMs to better understand relation definitions. (3) Mention boundary learning improves the LLM's ability to identify the boundaries of head and tail entities in triples.
- Extensive experiments demonstrate that the SDB-DRE model not only outperforms existing LLM baselines in terms of performance but also achieves better time efficiency due to its single-stage reasoning architecture, surpassing current methods in this regard.

2 Related Work

Document Level Relation Extraction. Most existing studies(Xiao et al., 2022; Lu et al., 2023; Jain et al., 2024; Gao et al., 2024) on DocRE rely on pre-labeled entities, which undermines their robustness when they deal with real-world scenarios (Meng et al., 2024). Some works (Eberts and Ulges, 2021; Xu and Choi, 2022; Zhang et al., 2023) shift towards exploring joint extraction methods for triplets from documents without relying on predefined entities. These approaches focus on mention detection and coreference resolution within the triplet extraction process. While refining the

evaluation metrics, they also introduce additional sources of cumulative errors and exposure bias in relation extraction. The performance advances of LLMs, along with the limitations of existing methods, motivate us to explore an end-to-end DocRE model that does not rely on pre-labeled entities.

LLMs and Relation Extraction. Existing LLM-based RE models (Wang et al., 2023; Xu et al., 2024) typically perform the RE task in a Question-Answering (QA) format. A common approach is to include a list of relation types in the model's prompt template as options. However, this setup is impractical in the DocRE scenario (Wadhwa et al., 2023), where it's not feasible to explicitly define the scope of relation types and their detailed descriptions in the input. For instance, Re-DocRED (Tan et al., 2022b) involves 96 relation types. Recent work (Xue et al., 2024) decomposes document-level triple extraction into three steps: relation identification, head entity identification, and the extraction of entity-relation triples. It provides an LLM-based method that integrates relation definitions. However, the multi-step training and inference process introduces additional time overhead.

Despite these advancements, we identify significant room for improvement in single-stage LLMs, which still struggle with issues such as Document Structure Parsing Error, Relation Definition Ambiguity, and Entity Boundary Recognition Errors. These challenges motivate our research and the development of a direct LLM-based method for triple extraction from documents. Our approach effectively integrates relation definitions into memory, addressing several inherent issues in current single-stage paradigms. Additionally, it achieves superior performance and efficiency compared to multi-step methods.

3 Methodology

Figure 2 illustrates the overall architecture of SDB-DRE. First, we describe the fine-tuning process that equips the LLM with document parsing capability. Next, we introduce mention boundary learning and relation definition learning in the relation extraction training process. Finally, we outline the model's overall training objective and inference process.

3.1 Problem Fomulation

Given a document D consisting of n tokens, let the training set consist of documents and their corresponding triplet labels, represented as $D_T = \{D, L_r\}$, where L_r denotes the all triplet-based answer label, and l_r is the length of the label. Each entity e in the document may appear multiple times and have multiple aliases, referred to as mentions m. Unlike prior work on DocRE (Lu et al., 2023; Gao et al., 2024), our model does not rely on pre-labeled entities. Our goal is to use LLMs to directly generate all triplets T contained within the document in an end-to-end manner. Formally, we denote the LLM as f_{LLM}, with QA task's input instruction represented as I. The output token sequence generated by the model is denoted as: $Y = f_{LLM}(I, D) = \{y_i\}_{i=1}^{l_r}$, where Y represents the answer generated by the model based on the input instruction and document content. The sampling probability of the tokens in Y is given by: $P(Y|I, D) = \prod_{t=1}^{l_r} P(y_t|I, D, y_{<t})$, where y_t denotes the t-th token in Y, and $y_{<t}$ represents the tokens generated before y_t.

3.2 Document Structure Awareness Training

To reduce reasoning errors in triples caused by document structure parsing mistakes, we perform pre-finetuning to enhance the document structure comprehension abilities of LLMs. Specifically, we define a QA task to help LLMs learn implicit document structure closely related to relation extraction: mention coreference and entity types. The former is crucial for reasoning about relationships between entity pairs across sentences, while the latter involves a potential logical connection between specific relation categories Formally, we construct document structure QA pairs based on the annotations available in the training documents. The input instruction for the document structure QA is denoted as I_s. The generated output token sequence is $Y_s = f_{LLM}(I_s, D) = \{y_s^i\}_{i=1}^{L}$, where Y_s represents the answers generated by the LLM based on the input instruction and document content, including the co-reference set of entity mentions and the corresponding entity types. The specific format can be found in Appendix F. We fine-tune our model with Parameter-Eficient method QLoRA (Dettmers et al., 2023). The objective loss function used for training is defined as follows:

$$\ell_s = -\sum_{(D,L_s)\in D_T} \sum_{t=1}^{l_s} \log P_{\theta+\theta_s}(y_t|I_s, D, y_{<t}) \quad (1)$$

where θ represents the parameters of the LLMs, θ_s denotes the parameters of QLoRA, and l_s is the length of the document structure label L_s. In this training step, we only update the parameters θ_s of the Structure-Aware QLoRA.

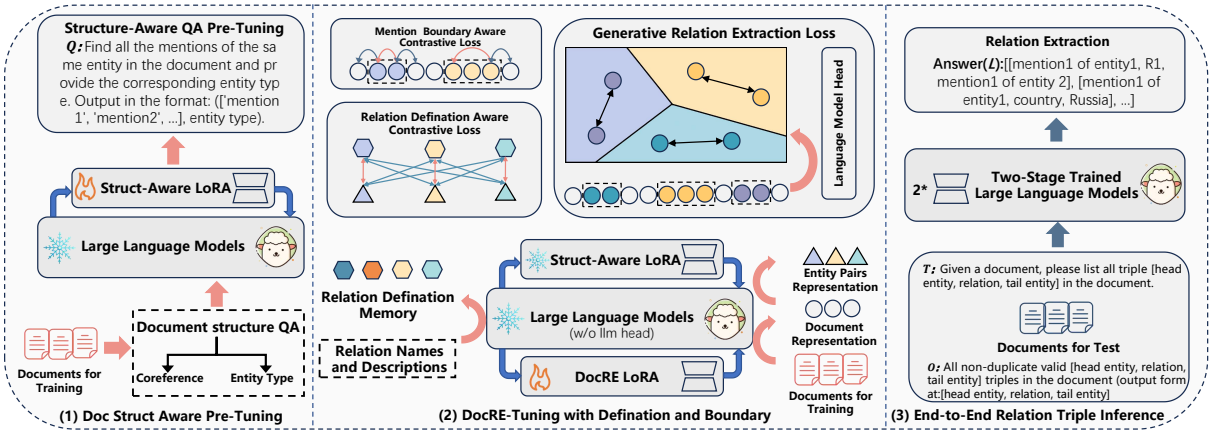

Figure 2: Overview of SDB-DRE. The red arrows indicate the training process and blue arrows indicate the inference process. The training process consists of two stages. The first stage helps LLMs acquire document structure parsing capabilities, while the second stage aim to alleviate errors in triples caused by inaccurate entity boundaries and conflicts between relationship definitions and triples.

3.3 Document Representation

Different from traditional generative RE training, our approach focuses on improving the document vector representations seen by the LLM head, thereby addressing the issues present in the existing single-step paradigm. Given a document D, we use the final hidden layer representation of the LLM corresponding to the document portion as the input document representation H. Following previous methods (Zhou et al., 2021), we extract the mention representation h_m from the positions in the document representation. By aggregating the representations of mentions belonging to the same entity, we obtain the entity representation: $h_e = \log\left(\sum_{i=1}^{|M_e|} \exp(h_{m_i})\right)$. Given an entity pair (e_h, e_t), we obtain the context representation associated with the entity pair through the attention matrix and the token representations of the entire document H:

$$c_{h,t} = H^T \frac{A_h \odot A_t^T}{A_h A_t} \tag{2}$$

It is worth noting that, due to the decoder-only architecture of the LLM, the document representation encoded by the model cannot access downstream information. This is detrimental to the model's ability to obtain sufficiently informative context representations for entity pairs. To address this, we unmask the document portion of the causal mask when computing the attention matrices A_h and A_t, ensuring that the model has visibility to downstream context information when calculating the key contextual representations for entity pairs. Notably, to avoid relying on pre-labeled entities for inference, the encoding of critical entity context information is only conducted during the training phase with labeled LLMs.

3.4 Relation Defination Learning

For each relation r, we use LLMs to encode the rewritten relation names and their corresponding definitions from prior work. We place the generated special start token of each LLM at the end of the document to capture the overall semantic meaning, and use the vector representation of this token from the final hidden layer before the LM head as the representation of the relation type r. By leveraging the enhanced descriptive text for relation types and non-discrete relation labels, we can fully exploit the powerful representational capacity of pre-trained LLMs to generate reliable semantic vector representations of relations. This collection of static vectors is referred to as the "relation definition memory" of the LLMs. For performance and computational resource reasons, this memory is not updated during model training.

We adopt a sampling strategy that helps the LLMs better distinguish between similar relation categories. Specifically, We define the positive sample pair set as: $P = \{(c_{h,t}, r_p) : (h, r_p, t) \in T\}$ where the context representation of the entity pair and its corresponding relation defination memory is considered a positive sample pair. Since a candidate entity pair may correspond to multiple relations, it may be associated with multiple positive pairs. For negative samples selection, we first filter out all entity pairs that do not have any relations. Then, we calculate the similarity between relations based on the LLM's relation description memory,

selecting the $topk$ most similar relations for each category as candidates for negative samples:

$$r_n = topk(\frac{r_p \cdot r_o}{\|r_p\|\|r_o\|}) \quad (3)$$

where o represents all relations other than r. The $topk$ function returns the top k relations r_o most similar to r_p, which are then selected as negative samples r_n. For each entity pair, we construct the negative sample set as: $N = \{(c_{h,t}, r_n) : (h, r_n, t) \notin T\}$ using the context representations of the entity pairs in the positive sample set and the similar relations.

We design a loss function that encourages LLMs to mine documents from a semantic perspective based on the entity pair context and relationship type definitions. Intuitively, by utilizing the final hidden layer representation and attention mechanism of LLMs, we obtain the contextual representation associated with the entity pair. The contextual representation of an entity pair should be closer to the representation of the specific description of the relationship it possesses, while being pushed farther apart from the representations of relationships it does not possess. Specifically, let the set of all candidate entity pairs in document D be E:

$$\ell_{rd} = -\sum_{(h,t)\in E} \frac{1}{|P|} \sum_{(h,r_p,t)\in P} \log \frac{u(c_{(h,t)}, r_p)}{\sum_{(h,r_n,t)\in N} u(c_{(h,t)}, r_n)} \quad (4)$$

where $u(c_{(h,t)}, r) = exp(sim(c_{(h,t)}, r)/\pi_{er})$ and π_{er} is a temperature hyperparameter. This loss function mitigates interference caused by the absence of the definition of relationship and further aids the model in distinguishing between relationship categories with similar semantics.

3.5 Mention Boundary Learning

We design a mention boundary learning loss to alleviate the errors in head-tail entity boundary identification within triples of LLMs. Specifically, let M denote the set of all mentions of entities in the document D, and m as a single mention of an entity. We construct positive and negative sample pairs based on the entity boundary tokens. the positive sample set is defined as: $P = \{(m_b, m_{e-1}) : m \in M\}$, where b is the start index of the mention and $e-1$ is the end index of the mention. In contrast, the negative sample set are defined as $N = \{(m_b, m_{b-1}), (m_{e-1}, m_e) : m \in M\}$ By utilizing a fixed ratio of positive to negative samples, we design a loss function that helps the model better identify mention boundaries, thereby improving the accuracy of triples. Intuitively, the token representations of mention boundaries should be pushed apart from those of the surrounding external tokens, while the start and end tokens of an entity should be brought closer together. Then the loss is defined as:

$$\ell_{mb} = \frac{1}{|M|} \sum_{m\in M} \log \frac{u(m_b, m_{e-1})}{u(m_b, m_{b-1}) + u(m_{e-1}, m_e)} \quad (5)$$

where $u(m_b, m_{e-1}) = exp(sim(c_{(h,t)}, r)/\pi_{mb})$ and π_{mb} is the temperature hyperparameter. This loss function mitigates the issue of fuzzy boundaries in generated triples and further helps LLMs distinguish entity mention boundaries within document representations.

3.6 Training objectives and inference

Given a document D and the input instruction for relation extraction QA represented as I_r, we use autoregressive generation loss ℓ_{re} to train the model for the final goal of relation extraction:

$$\ell_{re} = -\sum_{(D,L_r)\in D_T} \sum_{t=1}^{l_r} \log P_{\theta+\theta_s+\theta_r}(y_t | I_r, D, y_{<t}) \quad (6)$$

where l_r denotes the length of the relation extraction label L_r, and θ_r represents the parameters of the relation extraction QLoRA. Combining entity definition contrastive learning and mention boundary learning, the final training objective for the relation extraction QA phase is formulated as follows:

$$\ell = \alpha\ell_{rd} + \beta\ell_{mb} + \ell_{re} \quad (7)$$

where α and β are adjustable hyperparameters. In this training step, the parameters θ of the LLM and the parameters θ_s of the structural QLoRA, which were pre-trained in the first stage, are frozen. The newly introduced relation extraction QLoRA parameters θ_r are trainable. After completing the two-stage training, we perform inference using the LLM combined with both QLoRA parameters. Specifically, this is formulated as:

$$P(Y|I, D) = \prod_{t=1}^{l_r} P_{\theta+\theta_s+\theta_r}(y_t | I, D, y_{<t}). \quad (8)$$

The generated tokens are sampled from probabilities P and decoding with a fixed output format, we obtain all the triples contained in the document. The specific format of the prompts for relation extraction training and inference can be found in Appendix G.

4 Experiments

4.1 Dataset and Metric

Dataset. We conduct our experiments using the Re-DocRED (Tan et al., 2022b) dataset, refined by previous works (Xue et al., 2024). This dataset includes a validation set with 498 articles and 17,236 triplets, ensuring comprehensive and precise evaluation. It also contains a test set with 499 articles and 17,448 triplet facts. The dataset originates from DocRED (Yao et al., 2019), and Re-DocRED addresses the issue of an excessive number of false negative samples in the original dataset. Building on this, AutoRE (Xue et al., 2024) further modifies the relation descriptions and performs data cleaning.

Metric. We adopt the evaluation metric from previous work on end-to-end triple extraction using LLMs (Xue et al., 2024), which is designed for scenarios that do not rely on pre-labeled entities. This metric follows a strict Micro F1 standard, where a prediction is considered correct only if it exactly matches the relation, as well as both the head and tail entities. Notably, in the Re-DocRED dataset, a single triple may contain multiple aliases (mentions) for both the head and tail entities. A prediction is considered correct as long as it identifies any valid triplet pair. If the predicted pair matches any alias pair for the head and tail entities, it is counted as correct. However, other valid aliases are not counted in the correct statistics, meaning each correct triplet is counted only once. In contrast, all incorrect predictions, including entity mentions and relations, are considered false positives. This approach ensures a rigorous and statistically valid evaluation, enhancing the credibility of the final results. The implement detals can be found in Appendx A.

4.2 Baselines

We compare the proposed SDB-DRE method with three categories of DocRE baseline methods: (1) joint extraction methods based on traditional PLMs, (2) LLMs with single-stage inference, and (3) LLMs with multi-stage inference. It is important to note that most existing relation extraction models are tested with pre-defined entities, making their performance non-comparable to the method proposed in this paper.

PLM Method. This category includes TABLEFILLER (Zhang et al., 2023) and the current state-of-the-art (SOTA) model for document-level joint extraction, TAG (Zhang et al., 2023), which is the first to report end-to-end relation extraction results on Re-DocRED.

AutoRE. AutoRE (Xue et al., 2024) consists of a series of relation extraction models based on large language models (LLMs), covering various reasoning paradigms. The $D-F$ paradigm directly extracts factual triples from the document. The $D-RS-F$ paradigm first identifies all relation types present in the document and then extracts the corresponding triples for each relation. The $D-R-F$ paradigm extracts triples sequentially for each identified relation. The three-stage $D-R-H-F$ paradigm further decomposes the process by first identifying relation types, then extracting the head entities, and finally generating the full triples. This framework systematically explores how different levels of reasoning granularity affect relation extraction performance.

In addition, we also compare the performance of ChatGPT4o[1] to further support the superiority of our method. Details can be found in Appendix H.

4.3 Main Results

The experimental results on the Re-DocRED dataset are presented in Table 1. The results show that our method outperforms all strong baselines and the current SOTA model, AutoRE. Compared to traditional PLM-based methods, our method improves **5.05** F1 on the dev set and **4.94** F1 on the test set. When using the same base LLM[2] (Jiang et al., 2023), our method outperforms the single-stage inference LLM model AutoRE$_{D-F}$ by **15.32** on the dev set and **14.67** on the test set. Furthermore, compared to the SOTA multi-stage inference LLM-based model AutoRE$_{D-R-H-F}$, our method improves by **1.38** on the dev set and **2.41** on the test set. These improvements demonstrate that our proposed two-stage training effectively alleviates typical errors found in single-stage inference LLMs. At the same time, our method achieves superior performance with lower time costs compared to multi-stage inference LLMs. To further explore the upper bound of our method, we replace the base LLM with a more advanced one, Llama3 (Dubey et al., 2024), and observe further performance improvements. This highlights the robustness and potential of our approach. However, our method

[1] openai.com/api. The version is gpt-4o-2024-11-20.
[2] The version we use is Mistral-7B-Instruct-v0.2.

Model/Micro F1(%)	Base LM	Dev			Test		
		Precision	Recall	F1	Precision	Recal	F1
PLM							
TABLEFILLER*	RoBERTa-base	-	-	48.35	-	-	48.94
TAG*	RoBERTa-base	-	-	49.34	-	-	49.38
LLMs (Multi-Stage)							
AutoRE$^{*}_{D-RS-F}$	Mistral-7B-Instruct	-	-	40.30	-	-	40.33
AutoRE$^{*}_{D-R-F}$	Mistral-7B-Instruct	-	-	42.52	-	-	41.48
AutoRE$^{*}_{D-R-H-F}$	Mistral-7B-Instruct	**66.60**	44.02	53.01	**66.24**	42.67	51.91
LLMs (Single-Stage)							
ChatGPT4o	-	17.53	7.24	10.25	17.66	7.27	10.29
AutoRE$^{*}_{D-F}$	Mistral-7B-Instruct	-	-	39.07	-	-	39.65
SDB-DRE$_{Mistral-7B-Instruct}$	Mistral-7B-Instruct	62.73	48.01	54.39	62.85	47.83	54.32
SDB-DRE$_{Llama-3-8B-Instruct}$	Llama-3-8B-Instruct	60.90	**49.98**	**54.91**	62.60	**49.41**	**55.23**

Table 1: Results on the Re-DocRED benchmark. Scores of existing methods marked with a * are from the previous paper (Zhang et al., 2023; Xue et al., 2024). The best-performing method's metric values are highlighted in bold.

Model/(%)	Precision	Recall	F1
SDB-DRE	**62.60**	49.41	**55.23**
w/o Entity Type	59.89	**49.94**	54.46
w/o Coreference	60.26	47.82	53.32
w/o Structure Aware QA	57.70	49.09	53.05
w/o RDCL	58.05	49.11	53.21
w/o MBCL	62.10	49.12	54.85
w/o Two-Stage Training	59.13	47.70	52.79

Table 2: Ablation experiment on the ReFREDo test set, with the base LLM Llama3-8B-Instruct.

also exhibits certain limitations. While it achieves an increase in F1 score compared to multi-stage inference method, it shows a decrease in precision. This suggests that multi-stage inference models are better at filtering out incorrect relations, but they may also weaken the integrity of correctly identified triplets.

4.4 Ablation Study

We conduct a comprehensive set of ablation experiments on the test set to evaluate the effectiveness of each component. The results are shown in Table 2. Below is a detailed analysis of each part:

w/o Entity Type. In this experiment, we remove the entity type determination task from the SAQA and instead have the LLM only identify mentions in the document and aggregate them into a set based on co-reference structure. The model's performance shows a noticeable decline, indicating that the latent logical relationships between triples and entity types are crucial for the LLM's reasoning about relationship types.

w/o Coreference. In this experiment,we remove the co-reference structure parsing from SAQA and modify the instructions and answers to focus on identifying all mentions and their corresponding entity types. The model's performance declines, showing that explicitly training the LLM to parse co-reference structures improves relationship extraction performance.

w/o Structure QA. In this experiment, we remove all the SAQA training and the corresponding modules from the first stage. The model's performance further deteriorates, demonstrating the necessity of the structural parsing ability of LLMs in DocRE.

w/o RDCL. In this experiment, we remove the Relation Definition Contrastive Loss from .The results show that learning the relationship definitions effectively prevents the model from making incorrect inferences based solely on the relationship names, thus significantly improving the LLM's performance.

w/o MBCL. We remove the Mention Boundary Contrastive Loss (MBCL). The performance drop observed here shows that this training is essential for the model to more accurately identify the boundaries of the head and tail entities in the triples, resulting in more precise extraction.

w/o Two-Stage Training. we retain both QA tasks and the two contrastive losses, using a sin-

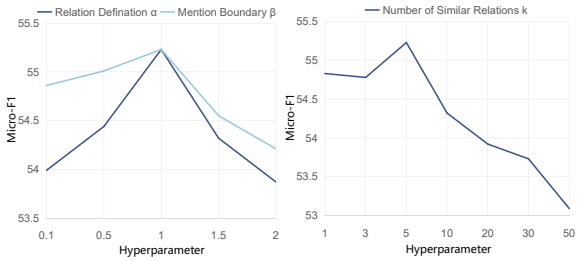

Figure 3: Experimental results of hyperparameters α, β and k. In single-parameter experiments, the remaining parameters are fixed at their estimated optimal values.

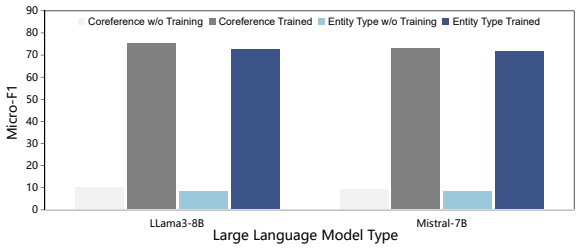

Figure 4: Performance Changes in Coreference Resolution and Entity Classification Before and After SAQA Training in LLM

gle QLoRA module for training. The model performance decreases, indicating that our two-stage training strategy has the following benefits: (1) avoids the impact of data distribution imbalance for different QA objectives on model performance; (2) allows the model to adapt to the frozen structure QLoRA parameters during the second-stage training; (3) preserves as much of the structural knowledge learned in the first stage during the second-stage training.

4.5 Analysis and Discussion

Hyperparameters Study. We investigate the impact of different hyperparameters on model performance by conducting experiments on the Re-DocRED test set. In the single-parameter experiments, we fix the remaining parameters to their estimated optimal values. As shown in Figure 3, the auxiliary coefficient α in relation definition learning and the auxiliary coefficient β in entity boundary learning play a crucial role in balancing the loss function. As these two parameters increase, the F1 score exhibits a clear trend of initially rising and then decreasing, with the optimal balance near 1.0. This suggests that teaching the LLMs to learn the relationship definitions and entity boundaries significantly improves the accuracy of triplets. However, when the balancing coefficients are too large, noise may be introduced due to label errors, missing data, and differences in objectives, leading to a decrease in triplet extraction performance.

Regarding the number k of similar relations sampled in the relationship definition learning's negative sampling, we observe that it has a minimal impact on overall performance. However, when the number is excessively large, it results in an imbalance between positive and negative samples in relationship definition learning, as well as an overabundance of similar relations. This hinders the model's ability to effectively distinguish between the definitions of easily confused relations, potentially leading to a negative impact on performance.

Document Parsing Ability Study. To investigate whether the model's ability to parse document structure genuinely improves after the first stage, we test the performance changes of the model before and after the SAQA training. Specifically, we define two evaluation metrics based on sets of entity mentions: (1) Coreference Resolution Metric: A set is considered correct only if all mentions of the entities in the set are classified correctly. (2) Entity Classification Metric: A set is considered correct only if the entire set is assigned the correct entity class. As shown in Figure 4, the experimental results indicate that after the first stage of SAQA training, the model's ability to parse document structure improves significantly. This lays a solid foundation for implicit document structure analysis and category-based logical reasoning during the relation extraction phase, effectively enhancing the performance of triplet extraction.

5 Conclusion

In this work, we propose an LLM-based DocRE method. First, we train the model to acquire document structure parsing capabilities through Structure-Aware QA. During the relationship extraction training, we introduce relation definition learning and mention boundary learning to mitigate the challenges that LLMs face when extracting triples. Our approach does not rely on pre-existing entity annotations during inference, making it more aligned with real-world application needs. Experimental results and further analysis demonstrate that our model outperforms existing methods on the public benchmark Re-DocRED, highlighting the superiority of our approach.

Limitations

Although our method adopts a novel two-stage training approach to enhance the performance of single-stage inference LLMs on DocRE, it shows a general decline in precision compared to multi-stage inference methods. In future work, we plan to introduce a more effective implicit reasoning process for LLMs to improve the accuracy of the generated triples. While SDB-DRE removes the reliance on pre-labeled entities during the inference phase, potential annotation errors during training can still impact the model's performance, motivating us to further explore and develop more robust methods. Additionally, SDB-DRE is limited to handling seen relation categories, which prompts us to develop methods (Popovic and Färber, 2022; Meng et al., 2023) with better generalization capabilities in the future.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: efficient finetuning of quantized llms. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pages 10088–10115.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Markus Eberts and Adrian Ulges. 2021. An end-to-end model for entity-level relation extraction using multi-instance learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3650–3660.

Chufan Gao, Xuan Wang, and Jimeng Sun. 2024. Ttmre: Memory-augmented document-level relation extraction. *arXiv preprint arXiv:2406.05906*.

Monika Jain, Raghava Mutharaju, Ramakanth Kavuluru, and Kuldeep Singh. 2024. Revisiting document-level relation extraction with context-guided link prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18327–18335.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Chonggang Lu, Richong Zhang, Kai Sun, Jaein Kim, Cunwang Zhang, and Yongyi Mao. 2023. Anaphor assisted document-level relation extraction. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Shiao Meng, Xuming Hu, Aiwei Liu, Fukun Ma, Yawen Yang, Shuang Li, and Lijie Wen. 2024. On the robustness of document-level relation extraction models to entity name variations. *arXiv preprint arXiv:2406.07444*.

Shiao Meng, Xuming Hu, Aiwei Liu, Fukun Ma, Yawen Yang, Lijie Wen, et al. 2023. Rapl: A relation-aware prototype learning approach for few-shot document-level relation extraction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5208–5226.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Nicholas Popovic and Michael Färber. 2022. Few-shot document-level relation extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5733–5746.

Qingyu Tan, Ruidan He, Lidong Bing, and Hwee Tou Ng. 2022a. Document-level relation extraction with adaptive focal loss and knowledge distillation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1672–1681.

Qingyu Tan, Lu Xu, Lidong Bing, Hwee Tou Ng, and Sharifah Mahani Aljunied. 2022b. Revisiting docred-addressing the false negative problem in relation extraction. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8472–8487.

Somin Wadhwa, Silvio Amir, and Byron C Wallace. 2023. Revisiting relation extraction in the era of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15566–15589.

Xiao Wang, Weikang Zhou, Can Zu, Han Xia, Tianze Chen, Yuansen Zhang, Rui Zheng, Junjie Ye, Qi Zhang, Tao Gui, et al. 2023. Instructuie: Multi-task instruction tuning for unified information extraction. *arXiv preprint arXiv:2304.08085*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz,

et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Yuxin Xiao, Zecheng Zhang, Yuning Mao, Carl Yang, and Jiawei Han. 2022. Sais: Supervising and augmenting intermediate steps for document-level relation extraction. In *2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022*, pages 2395–2409. Association for Computational Linguistics (ACL).

Yiqing Xie, Jiaming Shen, Sha Li, Yuning Mao, and Jiawei Han. 2022. Eider: Empowering document-level relation extraction with efficient evidence extraction and inference-stage fusion. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 257–268.

Jun Xu, Mengshu Sun, Zhiqiang Zhang, and Jun Zhou. 2024. Chatuie: Exploring chat-based unified information extraction using large language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 3146–3152.

Liyan Xu and Jinho D Choi. 2022. Modeling task interactions in document-level joint entity and relation extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5409–5416.

Lilong Xue, Dan Zhang, Yuxiao Dong, and Jie Tang. 2024. Autore: Document-level relation extraction with large language models. *arXiv preprint arXiv:2403.14888*.

Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. 2019. Docred: A large-scale document-level relation extraction dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 764–777.

Ruoyu Zhang, Yanzeng Li, and Lei Zou. 2023. A novel table-to-graph generation approach for document-level joint entity and relation extraction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10853–10865.

Wenxuan Zhou, Kevin Huang, Tengyu Ma, and Jing Huang. 2021. Document-level relation extraction with adaptive thresholding and localized context pooling. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 14612–14620.

A Implement Details

Our model is implemented using the PyTorch (Paszke et al., 2019) library and HuggingFace Transformers (Wolf et al., 2019). During the training of LLMs, we use a learning rate of 2e-5, a batch size of 1, and a maximum sequence length of 1024. The first 6% of the steps followed a linear warm-up, after which the learning rate decay linearly to 0. We perform early stopping based on the micro F1 score on the development set. All of our experiments are conducted on a single RTX 4090 GPU. In the parameter-efficient fine-tuning technique QLoRA, we follow the settings from prior work, setting the rank to 300 and the merge ratio to 16. The hyperparameters α, β, π_{rd}, π_{mb} and k were set to 1.0, 1.0, 0.5,0.5 and 5.0, respectively.

B Error Analysis

We randomly select 50 documents from the Re-DocRED test set and use a LLaMA3-8B-Instruct model trained with the single-stage method to generate predictions. Based on the gold labels, we extract all incorrect triplets and ask six annotators (divided into two groups) to label how frequently each of the three error types occurs. Each document is independently annotated by three annotators, and we report both the average and standard deviation to ensure annotation reliability. Additionally, we apply the same annotation process to the predictions generated by our proposed SDB-DRE model to further demonstrate its effectiveness in reducing these errors. The changes in the number of the three typical error types are shown in Figure 6. The changes in the proportion of the three typical error types within the total number of errors are shown in Figure 5. We can observe that: (1) These three types of errors are relatively common among the incorrect predictions. (2) Our method substantially reduces the frequency of these errors, both in terms of absolute count and their proportion among all incorrect predictions.

C Time Consumption

we measure the time required for both models to perform inference on the full Re-DocRED test set using the Mistral-7B-Instruct model on a single RTX 4090 GPU. To ensure fairness, no inference acceleration techniques were used. The results are Shown in Figure 7. We can observe that: our single-stage model significantly reduces inference time while also achieving better performance. Multi-stage methods require repeated document encoding and fine-grained decoding for each triple, which greatly increases computational cost.

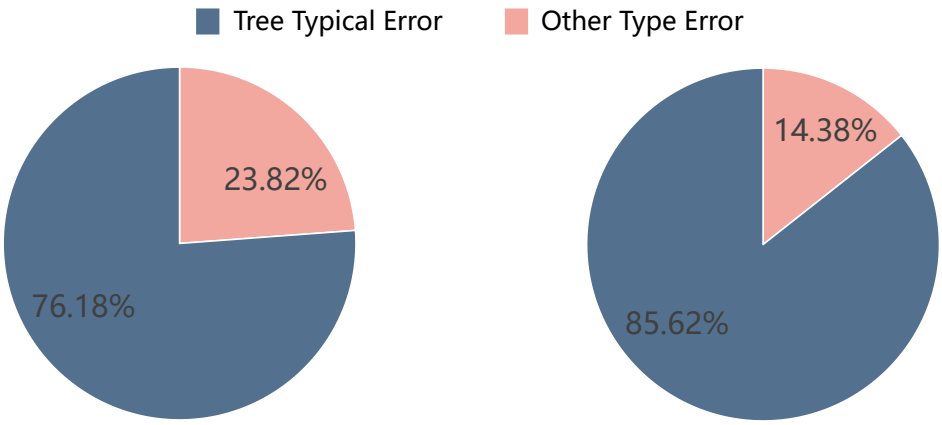

Figure 5: The proportion of the three typical error types within the total number of errors.

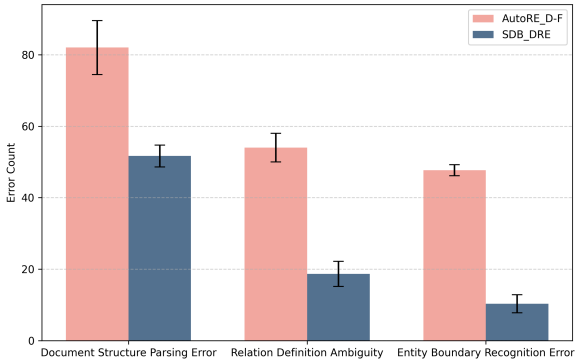

Figure 6: The variation in the number of three typical error types

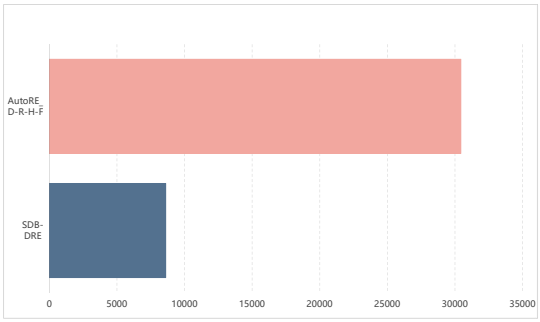

Figure 7: Time cost comparison between multi-stage and single-stage reasoning. The x-axis represents time, measured in seconds.

D Case Study

To more intuitively demonstrate the advantages of our proposed SDB-DRE over existing methods, we select representative documents from the ReDocRED test set for a case study. We showcase the answers of two single-stage LLM-based methods to visually highlight the superiority and limitations of each paradigm. As shown in Figure 8, compared to the previous single-stage reasoning model, AutoRE$_{D-F}$, we can find that:

- SDB-DRE correctly predicts the *country* relationship between *Kherson* and *Russian*. This shows that the SAQA training effectively enhances the document structural parsing capability of LLMs, enabling the model to derive this triple through further logical reasoning.

- SDB-DRE eliminates the incorrect prediction of [*Orlov, conflict, Nikolai Yudenich*] made by previous methods. This indicates that, with the learning of static relationship definitions in LLMs, the model better understands the specific semantics of different relationships, thereby mitigating the impact of relationship definition ambiguity on LLM performance.

- SDB-DRE correctly identifies the mention boundaries of *Soviet Naval Forces*, making the entire triple accurate. This shows that mention boundary learning effectively improves the LLM's ability to recognize mention boundaries in triples.

E Independent or Synergistic

To further examine the synergy among the losses, we carried out an experiment where we decouple the REQA training stage into two separate phases: one with contrastive learning only, and the other with REQA training alone. The comparison results are shown in the table 3. The model trained with contrastive learning and REQA in separate stages did not outperform the jointly trained model. This highlights the importance and effectiveness of the

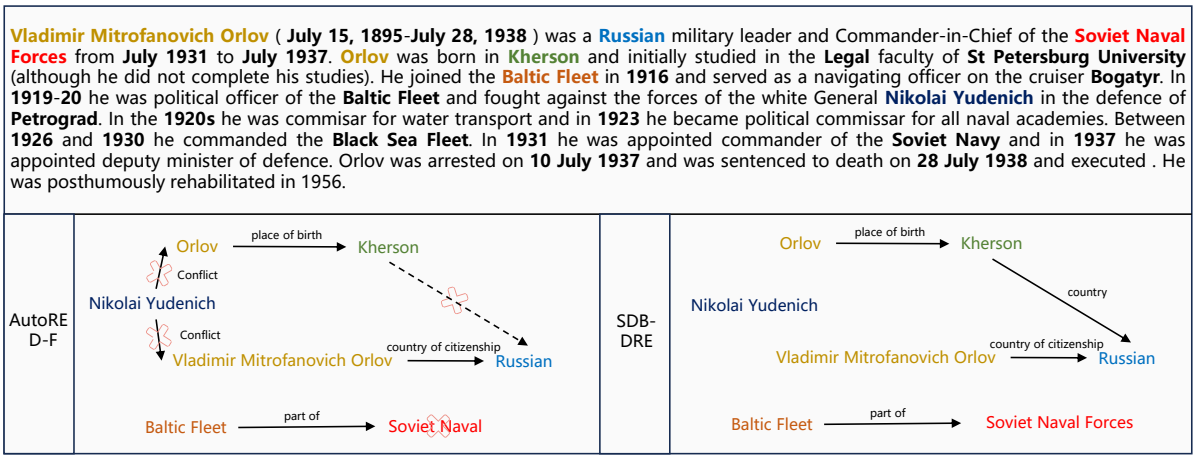

Figure 8: A case study of the single-stage reasoning methods AutoRE$_{D-F}$ and SDB-DRE. For the sake of brevity and ease of discussion, only representative prediction results are selected in the figure.

Model	Precisi on	Recall	F1
SDB-DRE$_{TwoStage}$	62.60	49.41	55.23
SDB-DRE$_{ThreeStage}$	60.02	49.45	54.30

Table 3: Training Results of Decoupled Contrastive Learning and Relation Extraction

synergistic interaction between loss terms, and validates our design choice of balancing these losses during joint training.

F Document Structure QA Prompt

The input prompts for the first-stage relation QA training are presented as follows:

Find all the mentions of the same entity in the document and provide the corresponding entity type. Output in the format: ({[mention1, mention2, ...], entity type)}.

Document: $Document$

All non-duplicate valid entity set, where each set includes all mentions of the entity in the document and the corresponding entity type. The output format is [mention1, mention2, ...], entity type.

The format of the label L_s in Structure-Aware QA during first-stage training is as follows:

[({mention1 of entity1, mention2 of entity1, ...}, entity1 type), ({mention1 of entity2, mention2 of entity1, ...}, entity2 type), ...]

G Relation Extraction QA Prompt

The input prompts for the second-stage relation QA training are presented as follows:

Given a document, please list all triple [head entity, relation, tail entity] in the document.

Document: $Document$

All non-duplicate valid [head entity, relation, tail entity] triples in the document (output format:[head entity, relation, tail entity], one triple per line, If there are no entities with existing relationships, return None):

The format of the label L in Relation QA during second-stage training is as follows:

[[mention1 of entity1, relation1, mention1 of entity2],[mention2 of entity1, relation1, mention1 of entity2],[mention1 of entity3, relation2, mention1 of entity2] .....]

H Detail of ChatGPT4o Test

The original ChatGPT model does not undergo task-specific instruction fine-tuning, which limits its familiarity with the target relational scope. To address this limitation, we change the baseline prompt design by explicitly incorporating the relational scope through name-based representations.

The input prompts for the relation QA are presented as follows: Given a document, please list all triple [head entity, relation, tail entity] in the document.

All candidate relation types are [relation1, relation2, relation3,...]

Document: $Document$

All non-duplicate valid [head entity, relation, tail entity] triples in the document (output format:[head entity, relation, tail entity], one triple per line, If there are no entities with existing relationships, return None):

In real-world settings, the variety of relationships is significantly more diverse, and incorporating their definitions into the prompt would substantially increase the computational cost of inference.