



ARTICLE

Enhancing Relational Triple Extraction in Specific Domains: Semantic Enhancement and Synergy of Large Language Models and Small Pre-Trained Language Models

Jiakai Li, Jianpeng Hu* and Geng Zhang

School of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai, 201620, China

*Corresponding Author: Jianpeng Hu. Email: mr@sues.edu.cn

Received: 24 January 2024 Accepted: 27 March 2024 Published: 15 May 2024

ABSTRACT

In the process of constructing domain-specific knowledge graphs, the task of relational triple extraction plays a critical role in transforming unstructured text into structured information. Existing relational triple extraction models face multiple challenges when processing domain-specific data, including insufficient utilization of semantic interaction information between entities and relations, difficulties in handling challenging samples, and the scarcity of domain-specific datasets. To address these issues, our study introduces three innovative components: Relation semantic enhancement, data augmentation, and a voting strategy, all designed to significantly improve the model's performance in tackling domain-specific relational triple extraction tasks. We first propose an innovative attention interaction module. This method significantly enhances the semantic interaction capabilities between entities and relations by integrating semantic information from relation labels. Second, we propose a voting strategy that effectively combines the strengths of large language models (LLMs) and fine-tuned small pre-trained language models (SLMs) to reevaluate challenging samples, thereby improving the model's adaptability in specific domains. Additionally, we explore the use of LLMs for data augmentation, aiming to generate domain-specific datasets to alleviate the scarcity of domain data. Experiments conducted on three domain-specific datasets demonstrate that our model outperforms existing comparative models in several aspects, with F1 scores exceeding the State of the Art models by 2%, 1.6%, and 0.6%, respectively, validating the effectiveness and generalizability of our approach.

KEYWORDS

Relational triple extraction; semantic interaction; large language models; data augmentation; specific domains

1 Introduction

Relational triples, essential to knowledge graphs, are formatted as (subject, relation, object) and are key to converting unstructured natural language texts into structured information [1]. In building domain-specific knowledge graphs, the extraction of these triples is critically important.

However, extracting relational triples for specific domains remains challenging. These tasks require highly accurate and domain-relevant triples. The uniqueness and complexity of texts in professional domains, such as extensive use of technical terms and specific semantic structures, make data processing in these areas particularly difficult [2]. Furthermore, the scarcity and difficulty in



obtaining specialized data limit the application of existing methods in these domains [3]. This situation calls for new approaches that not only make efficient use of limited data but also improve the model's adaptability and accuracy for domain-specific texts.

In our research, we have analyzed specific examples from domain-specific datasets like SciERC and CCL2022 to highlight the challenges in relational triple extraction. The SciERC dataset, consisting of scientific abstracts, includes triples like (“data structure”, “USED-FOR”, “phrase-based statistical machine translation”), demonstrating the nuanced relationships within academic discourse. Similarly, an example from the CCL2022 automotive industry fault dataset, “Before troubleshooting, there was a welding defect at the front beam welding point G101. The fault has been rectified.”, features the triple: (“front beam welding point”, “part_failure”, “welding defect”), reflecting the specific nature of automotive faults. In this context, ‘front beam welding point’ refers to a critical component in the structure of a vehicle, indicating a specific welding spot on the vehicle’s front frame. ‘Welding defect’ denotes issues that occur during the welding process, such as cracks or holes, which can compromise the structural integrity of the vehicle. ‘part_failure’ describes the malfunction of a component due to welding defects, potentially affecting the vehicle’s operation and safety. These examples highlight the challenges of extracting accurate triples from texts filled with technical jargon and complex sentence structures, further complicated by domain-specific contextual dependencies.

In the domain of relational triple extraction, initial studies mainly utilized a pipeline approach. A pipeline method refers to a technical approach where different processing steps are executed in sequence, such as first identifying entities in the text and then recognizing relationships between entities. For instance, the works of Zelenko et al. [4] and Chan et al. [5] are notable examples of this methodology, which faced challenges like error propagation and insufficient use of mutual information between entities and relations. Subsequently, there was a shift towards joint extraction methods, as exemplified by the works of Li et al. [6] and Zheng et al. [7]. Joint extraction methods process entities and their relationships simultaneously, aiming to reduce error propagation and increase accuracy. In recent years, the works of Xu et al. [8], Ren et al. [9] and Ning et al. [10] have set new standards in the field of relational triple extraction by integrating their innovative joint extraction methodologies.

Existing models have made progress in some areas, but they still face significant limitations in domain-specific triple extraction. Particularly, they often overlook the interdependencies and interactions between entities and relations [11,12]. In our study, we use a specially designed Attention Interaction Module, with key extensions integrating semantic information of relation labels, effectively enhancing text representation. This integration not only improves the interaction representation between entities and relations but also boosts the model’s performance in domain-specific triple extraction tasks.

In addition, large language models (LLMs) have shown their power in various Natural Language Processing (NLP) tasks, such as summary [13] and recommender systems [14], but their use in information extraction is limited by large data requirements and high training costs. Still, LLMs are uniquely advantageous in handling complex or ambiguous sentences [15]. Based on this, we propose a hybrid method using both large language models (LLMs) and fine-tuned small pre-trained language models (SLMs) specifically for reprocessing uncertain samples in domain triple extraction identified by SLMs. In our framework, SLMs handle initial sample classification and extraction, while LLMs focus on in-depth analysis and reevaluation of challenging samples identified by SLMs. This collaborative work allows our model to more accurately identify and process complex domain-specific triples, effectively overcoming the limitations of traditional methods.

To further leverage the potential of LLMs, our study harnesses their powerful capabilities in data augmentation to address the issue of domain data scarcity. We designed a novel process, utilizing LLMs to create a diverse training dataset by specifying personalized attributes like subdomains and context generated by LLMs. This method not only breaks the constraints of traditional data augmentation approaches but also provides broader and more varied data support for triple tasks, potentially enhancing model performance in specific domains.

Innovations in Relational Triple Extraction include: (1) We developed an Attention Interaction Module Enhanced by Relational Semantics, significantly enhancing semantic interaction efficiency between entities and between entities and relations by integrating relational semantic information. (2) We introduced a “voting strategy” for handling uncertain triples predicted by fine-tuned small pre-trained language models, leveraging large language models for reevaluation, which enhances prediction accuracy by utilizing their superior comprehension of complex contexts. (3) To address the scarcity of domain-specific data, we explored training data augmentation using large language models, creating a richer and more diverse dataset by introducing random attributes during the data augmentation process.

The remainder of this article is structured as follows: [Section 2](#) delves into related works, summarizing significant studies and their contributions to the field. [Section 3](#) begins with the problem definition of triple extraction, followed by an introduction to three key innovations designed to address these challenges. [Section 4](#) details the experimental setup, including outcomes, hyperparameter experiments, and analysis. Finally, [Section 5](#) concludes the article, summarizing the main findings and contributions.

2 Related Works

2.1 Relational Triple Extraction

The field of relational triple extraction has witnessed a significant evolution from rule-based methods to modern deep learning techniques. Recent research has focused on enhancing task performance and efficiency through various innovative strategies.

Early works, such as the Table Filling Multi-Task Recursive Neural Network (TF-MTRNN) model by Gupta et al. [16], utilized a tabular structure to model the interdependencies of entity recognition and relation classification, reducing the reliance on heuristic search methods. Effective as it was, this approach had computational efficiency limitations. Following this, Katiyar et al. [17] introduced an attention-based LSTM model that overcame the dependency tree limitation but faced challenges in handling large-scale, complex data. Zeng et al. [18] improved the handling of overlapping triples with their Seq2Seq model incorporating a copy mechanism, but it still had limitations in processing large-scale data and complex contexts.

With technological advancements, more innovative methods have been proposed to address complex issues in triple extraction. For example, the method by Wei et al. [19] and Li et al. [20] effectively handled multiple relation triple overlaps with a cascading binary framework. Wang et al. [21] innovatively transformed the extraction task into a token-pair linking problem, enhancing task performance. Studies by Zhong et al. [22] and Eberts et al. [23] improved model efficiency and implementation ease, using Transformer-based pretrained models for span-based extraction, further advancing the field.

Similar to our study’s approach, recent works have also focused on integrating different strategies to enhance triple extraction performance. For instance, the framework by Zheng et al. [24] guided

entity extraction by predicting potential relations, while the model by Wang et al. [25] processed entity detection and relation classification tasks simultaneously in a unified label space. Huang et al. [26] designed a pair-aware representation module and an entity-enhanced representation module to predict and combine directed entity pairs. Additionally, the model by Shang et al. [27] and the study by Tang et al. [28] focused on optimizing entity overlap handling and enhancing the accuracy of entity-relation interactions, respectively. Lastly, Yang et al. [29] incorporated label knowledge in entity extraction, achieving significant progress in enhancing text representation.

2.2 Applications of LLMs in Information Extraction

Recently, large pretrained language models (LLMs) like GPT-3 [30] and InstructGPT [31] have demonstrated exceptional performance in various downstream tasks, particularly in the information extraction (IE) domain. Studies by Martínez-Cruz et al. [32] and Sun et al. [33] verified the advanced capabilities of ChatGPT in keyword generation and information retrieval. Notably, Wei et al. [34] transformed zero-sample IE tasks into multi-turn Q&A problems with a two-stage framework. Similarly, Tang et al. [35] showcased LLMs' powerful ability to extract structured information from unstructured domain texts, highlighting their application prospects in domain-specific information extraction tasks. Chiang et al. [36] evaluated LLMs in content generation and editing tasks, finding their performance in text quality assessment comparable to human expert editors. Ma et al. [15] discovered that while LLMs are not efficient few-shot learners, they effectively complement SLMs in reprocessing difficult samples.

2.3 Applications of LLMs in Data Augmentation

As NLP continues to evolve, an increasing number of tasks and domains demand exploration. Many of these are resource-scarce and lack sufficient training examples, creating numerous vital applications that necessitate data augmentation [37]. Beyond their application in IE tasks, LLMs have also shown immense potential in data augmentation and annotation. Ding et al.'s study [38] explored the capabilities of GPT-3 as a data annotator, providing insights into GPT-3's role as a universal data annotator in NLP tasks. Yu et al. [39] focused on using LLMs as generators of training data with specific attributes, showcasing the potential of generating diverse and bias-reduced training data through LLMs. Additionally, Chung et al. [40] explored combining LLMs with human intervention to increase diversity in text data generation, achieving a balance between data quality and diversity while maintaining accuracy.

3 Proposed Method

3.1 Problem Definition of Triple Extraction

In the task of relational triple extraction, given a text sentence S , the objective is to identify all possible triples from S . Each triple can be formally represented as (E_1, R, E_2) , where E_1 and E_2 are entities within the sentence, and R is the relation connecting these two entities. Specifically, if we define sentence S as composed of a sequence of words w_1, w_2, \dots, w_n , where n is the length of the sentence, the goal is to find all sets of triples T that satisfy the following conditions: (1) Entities E_1 and E_2 are words or sequences of words within sentence S . (2) The relation R is one from a predefined set of relations, representing the semantic relationship between E_1 and E_2 . (3) Each triple (E_1, R, E_2) is semantically coherent and accurately reflects the meaning of sentence S .

The challenge of this task lies in accurately identifying and categorizing entities within the sentence and precisely determining the relationships between them. This becomes particularly difficult in

domain-specific texts due to the widespread use of specialized terminology and complex linguistic expressions.

3.2 Attention Interaction Module Enhanced by Relational Semantics

3.2.1 Encoder

In this research, we employ the BERT (Bidirectional Encoder Representations from Transformers) [41] pre-trained model as the core encoder to process and understand the input text. BERT is an advanced machine learning model that understands the context of language by learning from extensive text data. BERT utilizes a Transformer-based architecture to generate deep semantic representations through bidirectional contextual learning. Specifically, for a given input sentence S , we initially convert it into a series of tokens $T = \{t_1, t_2, \dots, t_n\}$, where n represents the number of tokens. This process employs the WordPiece tokenization method. Each token t_i is mapped to a high-dimensional vector space, producing embedding representations $\mathbf{E} = \{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n\}$. This means the model converts each word into a mathematical vector, capturing the word's meaning and its relationships with other words.

In our triple extraction task, relational labels are appended to the end of the input sentence as part of it and are inputted into the BERT model together. The BERT model processes these embeddings through a multi-layered Transformer network, each layer consisting of multiple self-attention heads to capture the interrelationships between tokens. For each token t_i within the sentence, the Transformer layer calculates an attention score \mathbf{A}_{ij} , indicating the relevance of t_i with all other tokens t_j . These attention scores are represented as:

$$\mathbf{A}_{ij} = \text{softmax} \left(\frac{\mathbf{Q}_i \mathbf{K}_j^T}{\sqrt{\dim_k}} \right) \quad (1)$$

Here, \mathbf{Q}_i and \mathbf{K}_j are the query and key vectors for tokens t_i and t_j , respectively, and \dim_k is the dimension of the key vectors. Through this mechanism, BERT captures the complex interplay of contextual relationships among tokens in the sentence, thereby facilitating a comprehensive and nuanced understanding of the text for effective triple extraction.

3.2.2 Relation Semantic Enhancement Module

In our model, the core of relation semantic enhancement is fortified through a specially designed attention interaction module, which strengthens the semantic linkage between the relation labels and their natural language descriptions. For this, we construct a natural language description D_0 for each relation type and obtain its representation \mathbf{D} through the BERT encoder. For instance, the description “Identify sentences describing malfunctions in product components, i.e., part failure. Such issues often involve hardware, software, elements, or other components of the product being faulty or damaged” serves as the natural language description for the “part_failure” relation. During the encoding phase, relation labels, along with the sentence, pass through the BERT model to obtain their embedding representation. We separate the embedding of the relation label part \mathbf{L} from the overall sentence embedding and input it into the relation semantic enhancement module alongside the embedding of the natural language description \mathbf{D} . The subsequent step involves mapping \mathbf{L} and \mathbf{D} to the same feature space and enhancing their semantic connection through an attention mechanism:

$$\mathbf{L}' = \mathbf{W}_1 \mathbf{L} + \mathbf{b}_1 \quad (2)$$

$$\mathbf{D}' = \mathbf{W}_2 \mathbf{D} + \mathbf{b}_2 \quad (3)$$

Here, $\mathbf{W}_1, \mathbf{W}_2$ are weight matrices, and $\mathbf{b}_1, \mathbf{b}_2$ are bias terms. Then, we compute the attention scores between each token $\mathbf{l}'_i \in \mathbf{L}'$ in the relation label \mathbf{L} and each token $\mathbf{d}'_j \in \mathbf{D}'$ in the relation description D_0 :

$$a_{ij} = \frac{\exp(\mathbf{l}'_i \cdot \mathbf{d}'_j)}{\sum_k \exp(\mathbf{l}'_i \cdot \mathbf{d}'_k)} \quad (4)$$

This score signifies the relevance between \mathbf{l}'_i and \mathbf{d}'_j . Subsequently, for each token in the relation label, we aggregate fine-grained features from the relation description:

$$\mathbf{c}_i = \sum_j a_{ij} \mathbf{d}'_j \quad (5)$$

Finally, these aggregated features are combined with the original token representations and processed through a nonlinear activation function to yield the enhanced relation label embeddings:

$$\mathbf{h}_i = \tanh(\mathbf{v}(\mathbf{l}'_i + \mathbf{c}_i) + \mathbf{b}) \quad (6)$$

Here, \mathbf{h}_i represents the embedding vector of the i^{th} token in the relation label after semantic enhancement. \mathbf{v} is a weight matrix, and \mathbf{b} is a bias term. Through this method, our model generates embeddings closely related to the semantics of each relation type, thereby enhancing the model's semantic understanding capabilities in processing triple extraction tasks.

By directly appending the relation labels to the end of the sentence and encoding them together, followed by relation semantic enhancement (as shown on the left side of Fig. 1), we capture not only the semantic links between entities but also the interactions between entities and their potential relations. This approach provides a solid foundation for our objective-to identify semantically valid connected entity pairs and differentiate their roles in relations.

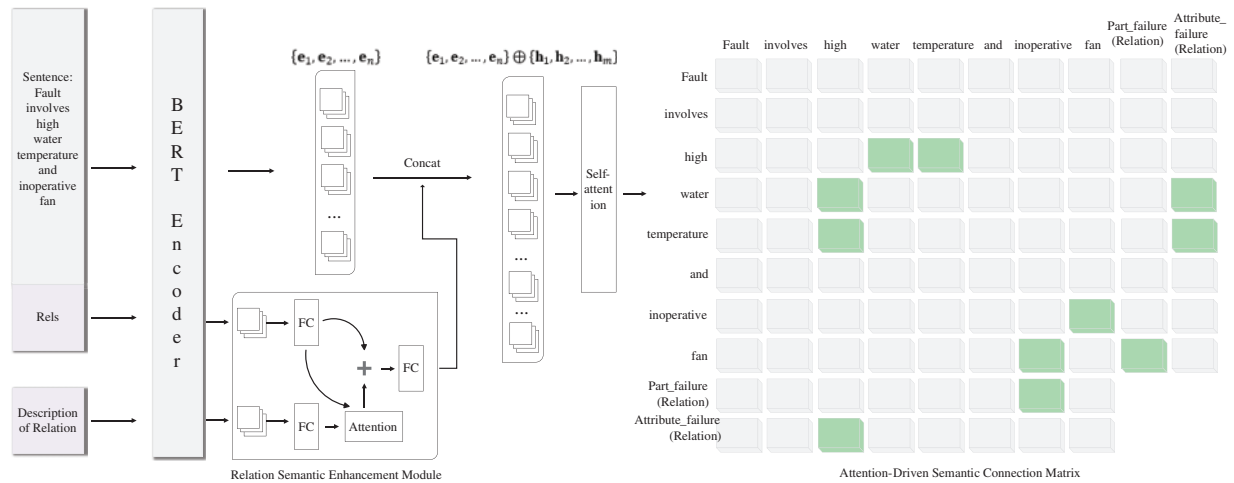


Figure 1: The structure of our model. The lower boxed area highlights the relation semantic enhancement module, while the matrix on the right side is the attention-driven semantic connection matrix. The green blocks within the matrix depict the model's predicted outcomes. Through the analysis of semantic connections between entities and relations, relational triples can be intuitively extracted from these results

3.2.3 Attention-Driven Semantic Connection Matrix

After the relation semantic enhancement module has been processed, we obtain the enhanced relation label embeddings \mathbf{h}_r . These embeddings are concatenated with the last hidden state embeddings \mathbf{e} , from the BERT model's output of the sentence, forming a unified representation. This concatenated representation \mathbf{H} is defined as:

$$\mathbf{H} = \{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n\} \oplus \{\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_m\} \quad (7)$$

Here, \oplus denotes the concatenation operation, $\{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n\}$ represents the embedding of the sentence, and $\{\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_m\}$ represents the enhanced embeddings of the relation labels. This merged representation \mathbf{H} encapsulates the sentence's inherent semantic information along with a deeper understanding of the relationship gleaned from the relation semantic enhancement module.

To capture these complex semantic connections, we defined an attention-driven semantic connection matrix M , which enables us to conduct an in-depth analysis of the semantic interactions between entities, potential semantic connections, and their roles within relations. The matrix M is further processed using a self-attention mechanism as follows:

$$M = \sigma \left(\frac{1}{H} \sum_{h=1}^H \frac{\mathbf{U}_h \mathbf{V}_h^T}{\sqrt{d_H}} \right) \quad (8)$$

In this formula, \mathbf{U}_h and \mathbf{V}_h respectively represent the embedding vectors derived from the entities and relations, H is the number of heads in the Transformer layer, and d_H is the dimensionality of each head. When the values of M exceed a specific threshold, we consider the corresponding semantic connection between the entity pair or between an entity and a relation to be valid.

The attention-driven semantic connection matrix M we construct is essentially based on the merged embedding representations of sentences and relation labels (as shown on the right side of Fig. 1). Specifically, each element M_{ij} of the matrix M reflects the strength of the semantic connection between the i^{th} element (possibly an entity within the sentence or part of a relation label) and the j^{th} element in the merged representation. In this manner, M can capture both the semantic connections between entities and the dynamic interactions between entities and relation labels.

3.2.4 Semantic Connections between Entities and Relations

Building upon the semantic connection matrix M introduced earlier, this study explores the intricate interactions between entity-entity pairs and entity-relation pairs within sentences as represented in matrix M . Our objective is to identify pairs of entities and their associated relations that can form semantically valid connections, analyzing these connections through the lens of matrix M .

We have defined a function F_{ee} to analyze whether two entities E_1 and E_2 in a sentence can form a meaningful semantic connection. This analysis is based on the hypothesis that if there exists a relation R that can connect these two entities, then their connection is meaningful. It is formalized as:

$$F_{ee}(E_1, E_2) = \exists R[(E_1, R, E_2) \vee (E_2, R, E_1)] \quad (9)$$

Here, E_1 and E_2 represent two entities in the sentence, and R represents a relation. If these two entities can be connected through relation R , then F_{ee} returns a truth value.

Another critical aspect is determining the role of an entity within a specific relation. We introduce the function F_{er} , which checks whether the entity E participates as a subject or object in relation R . The definition of this function is:

$$F_{er}(E, R) = \exists E [(E, R, _) \vee (_, R, E)] \quad (10)$$

where E is an entity, R is a relation, and the underscore ($_$) represents any entity. If there exists a valid triple with E as either a subject or an object, then F_{er} returns a truth value.

Through this comprehensive analysis, our model can not only understand the semantic connections between entities but also determine the roles of entities within specific relations. This provides a precise foundation for triple extraction.

3.2.5 Decoder

In this study, to train our model and assess the performance of relational triple extraction tasks, we employed the binary cross-entropy loss function. The formula is as follows [42]:

$$\mathcal{L} = -\frac{1}{(n+m)^2} \sum_{i=1}^{n+m} \sum_{j=1}^{n+m} (T_{ij} \log(M_{ij}) + (1 - T_{ij}) \log(1 - M_{ij})) \quad (11)$$

where T represents the actual label matrix of the semantic connection matrix M , indicating the true semantic connections between elements in the sentence. In this formula, n denotes the number of tokens in the sentence, m represents the number of relation types, M is the predicted semantic connection matrix, and T_{ij} and M_{ij} respectively are the elements at the i th row and j th column of the actual label matrix and the predicted matrix.

In our model, the decoding process extracts triples from sentences by utilizing the previously constructed attention matrix M as a scoring matrix. This matrix evaluates all possible entity pairs and relation labels for semantic consistency. Entity pairs that exceed a certain threshold score are selected as valid triples and combined with their corresponding relation labels. This process enables the model to effectively extract key structured information from sentences, enhancing the accuracy of triple extraction and the ability to handle complex sentences.

3.3 ‘Voting Strategy’ for Reevaluation

Traditional threshold methods are commonly employed in triple extraction tasks to decide whether to accept the model’s predicted outcomes. Such methods set a fixed threshold, for example, τ , and then compare each prediction’s confidence or probability p against it. If $p > \tau$, the prediction is considered definitive; otherwise, it is disregarded. However, this approach may not adequately address ambiguities in predictions, especially when the predicted values are close to the threshold. In such cases, even minor variations can lead to a change in the result from acceptance to rejection, or vice versa, a phenomenon particularly common with complex or ambiguous texts. Furthermore, a fixed threshold does not consider the complexity of the context and the specificity of the relations, which can be overly stringent or lenient in certain scenarios.

In response, LLMs such as GPT-3.5, with their deep semantic understanding and powerful contextual reasoning capabilities, offer an attractive solution. Trained on massive amounts of data, these models can capture subtle nuances and complex structures in language, making them ideally suited for evaluating and addressing uncertainties in predictions. By integrating LLMs into a “voting strategy,” we can reassess and confirm predictions made by SLMs that are near the threshold.

In our previous semantic enhancement attention interaction module, the model generated independent confidence scores for each potential triple, specifically targeting the head entity E_s , relation R , and tail entity E_o . These scores are respectively represented as confidence values, and we typically set a fixed threshold τ to determine whether to accept the model’s prediction. However, for predictions close to the threshold, the model’s judgment might be ambiguous, which we refer to as the confidence boundary interval.

Specifically, the confidence boundary interval refers to a small range $[\tau - \delta, \tau + \delta]$ around the threshold τ (The default setting is 0.5), where δ is a positive value representing the permitted uncertainty interval. When the score of any part of the triple $-s_{\text{head}}, s_R$, or s_{tail} falls within this interval, we consider the prediction of the triple to be uncertain. This is because they are close to the boundary determined by the model as valid or invalid, and minor changes or additional information might alter the final judgment.

In this study, we introduce a “voting strategy” based on LLM, specifically designed to address difficult samples predicted by SLM in triple extraction tasks. This strategy enhances prediction accuracy by having large models re-evaluate predictions close to the threshold, namely within the previously mentioned confidence boundary interval. Unlike traditional confidence assessment methods, we adopt a multiple-choice question format prompt to directly query the large model, thus leveraging its advantage in understanding complex contexts.

Specifically, for triples with uncertain relations, we design a series of carefully constructed questions, presenting them as multiple-choice questions, each revolving around a specific subject E_s and object E_o in the sentence. These questions aim to inquire about the most appropriate relation R to form a valid triple (E_s, R, E_o) . The template for the prompt is as follows, as shown in [Table 1](#).

Table 1: The template for the prompts

	Triples with uncertain relations	Triples with uncertain entities
Prompt	<p>Given a text from the domain of [insert domain] and sub-domain of [insert sub-domain]: “...”.</p> <p>When performing an NLP task for text analysis, please make a simple judgment on the subject: E_s, and the object: E_o in the following sentences. Please select the correct answer:</p> <p>A. Belongs to relation $R1$. B. Belongs to relation $R2$. C. Belongs to relation $R3$. D. None of the above.</p>	<p>Given a text from the domain of [insert domain] and sub-domain of [insert subdomain]: “...”.</p> <p>When performing an NLP task for text analysis, consider the subject: E_s and object: E_o in the sentence, as well as the potential triple sentence they form. For the following statement, please make a judgment: “E_o” is a type of (insert entity type) entity. “The relationship $E_s E_o$” belongs to relation “R”. Please select:</p> <p>A. The statement above is correct. B. There are incorrect elements in the statement above.</p>

In this template, E_s and E_o represent specific entities in the sentence, while $R1$, $R2$, and $R3$ represent possible relationship options. To assist the large model in making more accurate judgments, we also provide a simple explanation and rationale for each type of relationship in the prompt.

For triples with uncertain entities, we employ a similar method to design prompts, asking the large model for judgments about entity boundaries and types. The template for the prompt is as shown in [Table 1](#).

In these prompts, we leverage the capabilities of large models like GPT3.5 to parse and answer these questions, thereby obtaining definitive judgments about uncertain triples. Each returned option is equivalent to a vote, and we have set a specific threshold θ , when the number of votes for an option reaches θ (typically set at 5 votes), we accept that option as the final judgment. This method not

only fully utilizes the large model’s ability to understand complex texts and perform reasoning but also ensures the reliability and accuracy of the results by setting a threshold. The process flowchart is shown in Fig. 2.

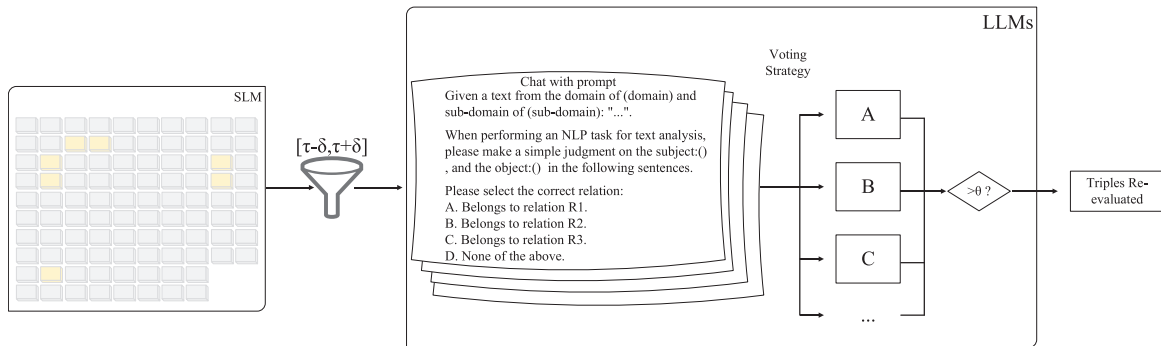


Figure 2: Flowchart of the ‘voting strategy’

Through this “voting strategy,” we can more effectively handle the uncertain predictions of SLM in triple extraction, and with the advanced semantic understanding capabilities of large-scale language models, we enhance the overall accuracy of predictions. The introduction of this strategy not only improves the performance of triple extraction but also provides a new perspective and tool for dealing with complex and ambiguous situations.

3.4 LLMs Driven Data Augmentation for Training

Data augmentation is a technique commonly used to enhance the performance of machine learning models by generating new training samples to expand the dataset. This approach is particularly useful in domains where data is scarce, helping the model better understand and handle the complexities of specific fields.

In this study, we address the prevalent issue of insufficient domain-specific data by proposing an innovative data augmentation approach. The scarcity of domain data often limits the performance of models in specific fields, especially in triple extraction tasks that require a large amount of fine-grained knowledge. To mitigate this deficiency, we harness the deep semantic understanding and generative capabilities of LLMs to construct richer and more diverse domain-specific datasets.

Our method begins by querying LLMs to identify the most representative and crucial elements or attributes for a specific domain. This is done to ensure our augmented datasets can cover the core topics and concepts within the domain. Taking scientific literature as an example, we first pose the following question to the LLMs: “What do you consider the most important elements or attributes for texts in scientific literature?” Such open-ended inquiries allow LLMs to return a series of high-quality and relevant attributes, such as background, objective, method, and results, based on their extensive knowledge and understanding.

Next, we delve deeper into each attribute by asking the LLMs for a range of possible values suitable for the dataset of each attribute. For instance, in the domain of car malfunctions, for the attribute of faulty components, LLMs might return options like “engine,” “exhaust pipe,” and “power system;” for the context of the malfunction, it might return scenarios like “during heavy rain” or “after driving through a muddy stretch.” To ensure that the generated data is both diverse and consistent with real-world domain specifics, we typically obtain no fewer than 200 possible values for each attribute.

In the process of delving into each attribute and its possible values, we particularly noted the strong correlation between certain attributes, a factor crucial for generating domain data that is both plausible and realistic. Take, for instance, the dataset of scientific papers, where the field and sub-field of a paper represent a pair of attributes with a strong correlation. In such cases, simple random combinations may lead to unrealistic or illogical attribute relationships, prompting us to adopt a specialized approach.

For attribute pairs with strong correlations, such as domain and sub-domain, we consider the sub-domain as a sub-attribute of the domain. This means that, rather than considering each attribute independently, we first determine the value of the primary attribute (like the domain) and then generate a series of related sub-attribute (such as sub-domain) values for each specific primary attribute value. This method ensures that in the subsequent combination process, the sub-domain always maintains a reasonable and consistent correlation with its respective domain.

For example, if 'Artificial Intelligence' is identified as a domain, its sub-domains might include 'Machine Learning,' 'Natural Language Processing,' and 'Computer Vision.' In this way, we ensure that each generated domain-sub-domain pair is inherently coherent and interrelated. This meticulous attribute generation strategy significantly reduces the irrationality caused by random combinations while enhancing the overall quality and practicality of the data augmentation process.

The final step is to augment the dataset by randomly combining these attributes and their values through templates to form prompts, akin to generating domain-specific "data scripts," each script being a possible simulation of real-world scenarios. Through this method, we can create a vast array of domain data with varying styles and scenarios, significantly expanding the coverage and diversity of the original dataset.

Fig. 3 is a specific example demonstrating how our data augmentation method can be used to expand the automotive industry's fault dataset. In the domain of automotive industry fault datasets, our method is particularly suitable for generating data with practical domain value and authenticity.

4 Experiment

4.1 Purpose of the Experiment

In this study, the experimental section is dedicated to thoroughly validating the effectiveness of our proposed approach in handling domain-specific relational triple extraction tasks. Specifically, the objectives of the experiments include validation of effectiveness: We aim to demonstrate that the relation semantic enhancement module, 'voting strategy,' and the data augmentation method proposed by us can significantly enhance the performance of models on specific domain datasets in relational triple extraction. This involves comparing our approach with existing techniques to showcase its advantages in dealing with scarce domain data, enhancing semantic understanding of relationships, and processing uncertainties.

4.2 Evaluation Metrics

To comprehensively evaluate the performance of our approach, we employ the following metrics:

Accuracy: This direct performance metric signifies the proportion of triples correctly extracted by the model. High accuracy means the model can accurately identify and categorize entities and relations in the text.

Recall: Recall measures the model's capability to identify all true triples. In relational triple extraction tasks, a high recall is particularly critical as missing key information could lead to incomplete or erroneous understanding of the text.

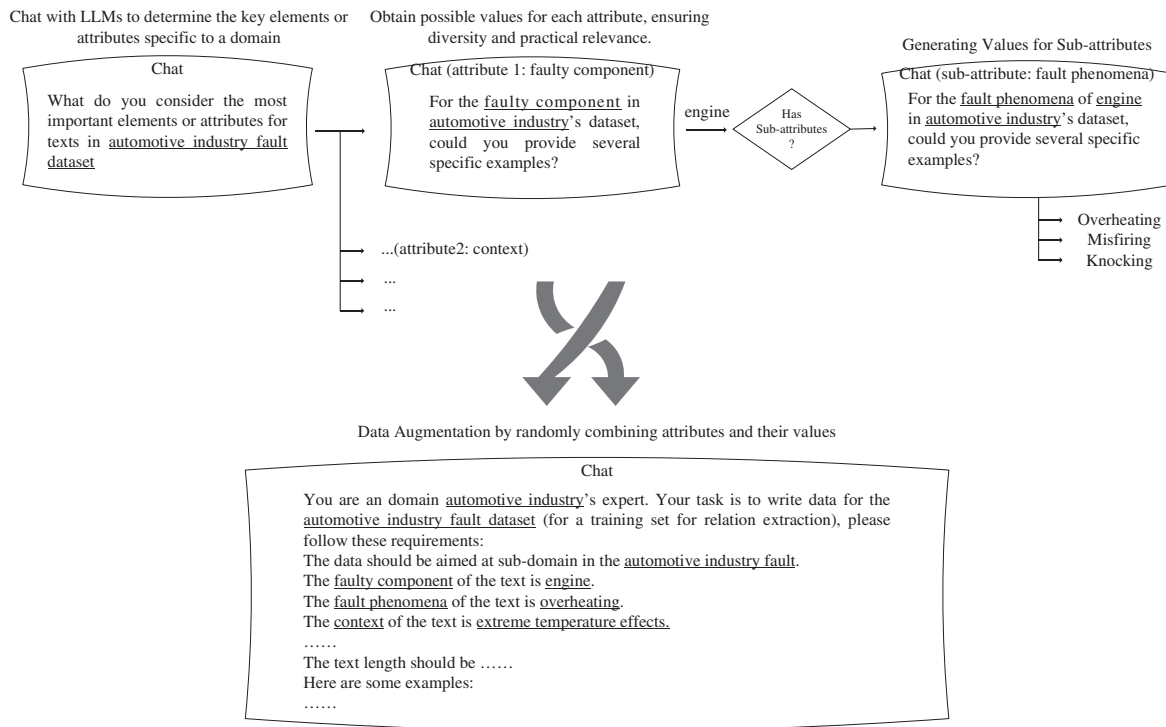


Figure 3: An example of data augmentation with flowchart

F1 score: This harmonic mean of precision and recall is a vital indicator for assessing the model’s accuracy and comprehensiveness. The F1 score provides a singular measure reflecting the model’s overall performance in precisely identifying triples while not omitting crucial information.

4.3 Datasets

In this study, to rigorously train our model and evaluate its performance in relational triple extraction tasks, we delve into a variety of datasets, each with its unique characteristics and challenges.

The contents of [Table 2](#) display the details of three datasets. The datasets we selected are characterized by the following features:

Table 2: Details of the datasets

Dataset name	Entity types	Relation types	Dataset size
CCL2022	3	2	3 k triples
SciERC	6	7	4.7 k triples
CoNLL04	4	5	2 k triples

(1) Limited Training Data: Particularly for domain-specific datasets, the limited amount of data presents challenges for the model’s learning and generalization capabilities. (2) Domain Datasets Contain Specialized Terminology: Domain-specific datasets often encompass an extensive range of

specialized terms and specific knowledge, increasing the difficulty for models to comprehend and process these terms.

4.3.1 CCL2022 Automotive Fault Domain Case Text dataset

This dataset consists of records written by repair professionals, detailing instances of car malfunctions, diagnostic steps, and resolutions. These records intricately describe the fault phenomena, the reasons behind the malfunctions, and the processes undertaken to rectify the issues.

4.3.2 SciERC Dataset

The SciERC [43] dataset is a collection of 500 scientific abstracts extracted from conference/workshop papers across 12 AI domains, sourced from the Semantic Scholar Corpus. These abstracts are annotated with scientific entities, their relations, and coreference chains. SciERC builds upon prior scientific paper datasets from SemEval 2017 Task 10 and SemEval 2018 Task 7 by expanding entity and relation types, enhancing relation coverage, and incorporating cross-sentence relations.

4.3.3 CoNLL04 Dataset

The CoNLL04 [44] dataset comprises news articles from The Wall Street Journal and the Associated Press. It defines four entity types and five relation categories, covering a variety of news reporting scenarios.

4.4 Baseline Models

CasRel [19]: Introduced a cascading binary tagging framework that naturally addresses the overlapping issue by considering relations as functions mapping subjects to objects. It has demonstrated outstanding performance across multiple datasets.

TPLinker [21]: As a one-stage joint extraction model, it incorporates a handshaking tagging scheme effectively discovering overlapping relations sharing one or two entities, immune to exposure bias issues.

Spert [23]: Presented an attention-based span-style joint entity and relation extraction model that efficiently performs entity recognition and relation classification through lightweight inference and robust negative sample training.

PURE [22]: Utilizes a straightforward pipeline approach for entity and relation extraction, establishing new best practices on standard benchmarks and showing advantages over shared-representation multitask learning approaches.

OneRel [27]: Proposes treating joint extraction as a fine-grained triple classification problem, effectively addressing cascading errors and information redundancy through score-based classifiers and relation-specific tagging strategies.

UniRel [28]: Enhances extraction effects while boosting computational efficiency by unifying the representations of entities and relations and constructing interaction graphs to comprehensively capture the rich associations between entities and between entities and relations.

PL-marker [45]: Introduces a novel span representation method considering the interrelationships between spans (pairs) and designs a subject-oriented packing strategy for complex span-pair classification tasks to better simulate the interrelationships among the same subject's span pairs.

TriMF [46]: The paper presents a framework that uses a memory flow attention mechanism and a trigger sensor to improve entity and relation extraction by enhancing bidirectional interaction and relation trigger detection.

PRGC [24]: The paper proposes the PRGC framework, which decomposes entity and relation extraction into three subtasks to enhance accuracy and efficiency by focusing on potential relations and solving overlap and alignment challenges.

EmRel [8]: This paper introduces a novel framework for relation extraction that enhances entity and relation representation through an attention-based fusion module and tucker decomposition, demonstrating superior multi-triple extraction performance.

BiRTE [9]: This paper introduces a bidirectional extraction framework with a shared-aware learning mechanism, significantly enhancing relational triple extraction accuracy.

OD-RTE [10]: The study presents a one-stage object detection framework for relational triple extraction, enhancing efficiency and accuracy through innovative decoding and negative sampling strategies.

4.5 Parameter Configuration

We set the following unified parameter configurations, as detailed in [Table 3](#).

Table 3: Parameter configuration

Parameter name	Value	Description
Learning rate	5e-5	Ensuring stable learning throughout the training process
Batch size	16	Balancing computational resource efficiency and gradient estimation accuracy
Number of epochs	100	Providing sufficient iterations for the model to thoroughly learn and converge
Length of natural language description of relation type	50	Ensuring it contains ample information
Confidence boundary interval (δ)	0.2	Determining the sensitivity for identifying uncertain predictions
Data augmentation multiplier	1	Controlling the scale and diversity of data augmentation

In [Table 3](#), the last three items are hyperparameters. In the subsequent experimental section, we will elaborate on the selection of these hyperparameters and their impact on model performance. Through these settings, we aim to comprehensively evaluate the performance of different models in handling specific tasks and validate the effectiveness and superiority of our proposed approach.

4.6 Results

In our experiments, all pre-trained models utilized the bert-base. For the Voting Strategy and Data Augmentation parts, the LLM employed was GPT-3.5. Here is a performance comparison of various models on different datasets, represented by Precision (P), Recall (R), and F1 score (F1), as shown in [Tables 4 and 5](#).

Table 4: Experimental results of classical models (before 2022)

Model	CCL2022			SciERC			CoNLL04		
	P%	R%	F1%	P%	R%	F1%	P%	R%	F1%
Casrel	69.3	55.0	61.3	41.9	41.3	41.6	69.9	69.4	69.6
TPlinker	62.6	63.3	63.0	42.6	42.9	42.8	66.6	67.2	66.9
Spert	65.2	63.5	64.3	49.8	43.5	46.4	74.8	71.5	72.9
PURE	67.6	62.1	64.7	49.0	47.9	48.4	70.2	69.2	69.7
TriMF	66.8	65.9	66.3	51.8	50.7	51.2	73.0	71.6	72.3
PRGC	66.7	65.8	66.2	49.1	47.5	48.3	69.0	68.4	68.7
Ours	69.3	67.8	68.6	54.1	52.1	53.1	74.3	72.8	73.5

Table 5: Experimental results of new models (2022 and 2023)

Model	CCL2022			SciERC			CoNLL04		
	P%	R%	F1%	P%	R%	F1%	P%	R%	F1%
Onerel	66.3	65.6	65.9	46.1	46.8	46.4	69.0	68.1	68.6
Unirel	66.6	63.9	65.2	48.5	48.2	48.4	69.8	69.4	69.6
PL-marker	–	–	–	–	–	50.8	–	–	71.3
EmRel	65.1	63.9	64.5	48.0	46.6	47.3	71.3	70.2	70.7
BiRTE	66.4	65.2	65.8	49.7	50.1	49.9	70.7	70.0	70.4
OD-RTE	67.2	66.0	66.6	51.8	51.2	51.5	72.9	72.3	72.6
Ours	69.3	67.8	68.6	54.1	52.1	53.1	74.3	72.8	73.5

The experimental results from three distinct datasets (CCL2022 Automotive Fault Domain, SciERC Scientific Literature, and CoNLL04 News Articles) showcase the comparative analysis of our method against other state-of-the-art models. The results indicate that our approach demonstrates significant performance advantages in multiple aspects.

On the CCL2022 Automotive Fault Domain dataset: Our method (“Ours”) exhibits robust performance in Precision (P), Recall (R), and F1 score, reaching 69.31%, 67.83%, and 68.56%, respectively. This indicates that our approach effectively captures domain-specific fault phenomena, causes, and processes, maintaining high recognition accuracy even amidst specialized terminology and complex contexts. Compared to other models, our method stands out in balancing accuracy and comprehensiveness, thanks to our unique data augmentation strategy and the “voting strategy” for uncertain predictions.

On the SciERC Scientific Literature dataset: When dealing with scientific literature abstracts that contain a multitude of entity and relation types, our approach also demonstrates superiority, reflected in a 54.13% Precision, 52.06% Recall, and 53.07% F1 score. This result highlights our method’s capability in handling complex, cross-sentence, and coreference relations. Particularly, compared to other models, our approach better comprehends the deep semantic relationships and terminologies in scientific literature, credited to our strategy of employing LLM for deep semantic understanding and attribute association.

On the CoNLL04 News Articles dataset: Our method also achieved notable results on the CoNLL04 dataset, attaining a Precision of 74.34%, a Recall of 72.75%, and an F1 score of 73.53%. This demonstrates the effectiveness of our approach in handling the specific domain of news, particularly in dealing with its diverse text structures. Compared to other models, our method more accurately extracts relational triples across various news texts.

4.7 Ablation Experiments

We conducted ablation experiments by individually adding a module to the baseline model (w/) or removing a module from the complete model (w/o). The results of these ablation studies are presented in [Table 6](#).

Table 6: Results of the ablation study. In the table, ‘RSEM’ stands for ‘relation semantic enhancement module’, ‘VS’ represents ‘voting strategy’, and ‘DA’ denotes ‘data augmentation’

Model	CCL2022			SciERC		
	P%	R%	F1%	P%	R%	F1%
Baseline model	66.0	62.7	64.3	46.2	44.1	45.1
w/RSEM	66.3	64.3	65.3	48.9	46.8	47.8
w/VS	68.7	64.4	66.5	50.5	48.4	49.4
w/DA	67.3	63.8	65.5	49.6	46.4	47.9
w/o RSEM	69.6	66.3	67.9	52.1	48.9	50.4
w/o VS	68.2	64.3	66.2	49.5	48.9	49.2
w/o DA	69.8	64.3	67.0	51.6	49.8	50.1
Complete model	69.3	67.8	68.6	54.1	52.1	53.1

Following detailed ablation experiments, we arrived at the following conclusions: Firstly, the introduction of the Relation Semantic Enhancement Module proved critically important in enhancing the semantic interaction efficiency both between entities and between entities and relations. On the CCL2022 dataset, the inclusion of this module improved the F1 score from 64.3% in the baseline model to 65.3% (+1.0%), while on the SciERC dataset, the improvement was even more significant, rising from 45.1% to 47.8% (+2.7%). Secondly, the voting strategy we proposed effectively enhanced the model’s capabilities, especially on the CCL2022 dataset, where the F1 score increased from 64.3% in the baseline model to 66.5% (+2.2%). Additionally, our data augmentation strategy, while significantly improving model performance with F1 score increases of +1.2% and +2.8% on the two datasets respectively, played a positive role in alleviating the issue of domain data scarcity.

We further validated the unique contributions of each innovation by comparing the performance of the complete model with and without individual innovative components. Notably, the removal of the voting strategy led to a significant decrease in model performance, with F1 scores dropping by 2.4% and 3.9% on the CCL2022 and SciERC datasets, respectively. This highlights the importance of these components in enhancing the model’s accuracy and robustness. Ultimately, the complete model outperformed all other configurations on the CCL2022 and SciERC datasets, with F1 scores reaching 68.6% and 53.1%, respectively. This further demonstrates the effectiveness of our approach in enhancing the performance of domain-specific relational triple extraction tasks.

4.8 Hyperparameter Experiment Analysis

We conducted three sets of hyperparameter experiments on the CCL2022 dataset to assess the impact of hyperparameters on model performance.

4.8.1 Hyperparameter Experiment 1: Length of Natural Language Description of Relation Types

In our hyperparameter experiment focused on the length of the natural language description of relation types (D_0), we investigated the impact of varying lengths on the model's performance, specifically in terms of Precision (P), Recall (R), and F1 score. The results are summarized as shown in [Table 7](#).

Table 7: Results of the hyperparameter experiment 1

Length	P%	R%	F1%
10	68.45	67.41	67.92
25	69.69	66.93	68.28
50	69.31	67.83	68.56
75	69.35	66.84	68.07

The [Table 8](#) below provides examples of natural language descriptions of relation “part_failure” at varying lengths.

Table 8: Examples of natural language descriptions of relation types at various lengths

Length	Description
10	Part error/failure, indicating malfunctions in product components.
25	Identify sentences with part failure, where part failure refers to malfunctions of product components.
50	Identify sentences describing malfunctions in product components, i.e., part failure. Such issues often involve hardware, software, elements, or other components of the product being faulty or damaged.
75	Identify sentences highlighting malfunctions in product components, termed part failure. This includes faults or damages in hardware, software, electrical elements, or other essential parts affecting the product's integrity, usability, or safety. Failures may present as operational inconsistencies, breakdowns, or subtle signs of component quality degradation, impacting the product's functionality and reliability.

The experiment indicates that the model achieved the optimal F1 score at a description length of 50 words, suggesting that a moderately detailed relation description aids the model in better understanding and extracting triples. Both shorter and longer descriptions appear to slightly diminish performance, potentially due to insufficient information or information overload, which could hinder the model's ability to capture key semantics effectively. Therefore, selecting an appropriate length for the description is crucial for optimizing model performance.

4.8.2 Hyperparameter Experiment 2: Confidence Boundary Interval

The size of the confidence boundary interval directly affects the model's range of handling uncertain predictions. We tested different interval sizes and observed their impact on the model's Precision (P), Recall (R), and F1 score, as presented in Table 9 and Fig. 4.

Table 9: Results of the hyperparameter experiment 2

The size of the confidence boundary interval	P%	R%	F1%
± 0.1	69.63	66.18	67.86
± 0.15	69.31	67.83	68.56
± 0.2	68.71	68.12	68.41
± 0.25	66.67	68.02	67.34

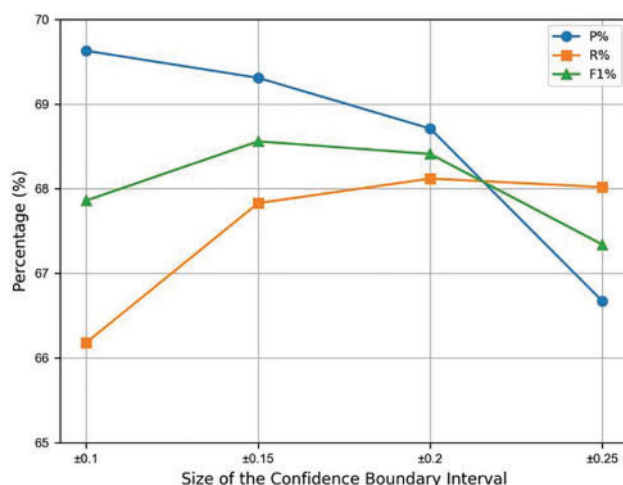


Figure 4: Results of the hyperparameter experiment 2

With the interval set to ± 0.1 , the model's F1 score peaked at 67.86%, suggesting a tighter interval may not encompass a sufficient number of uncertain predictions. This conservative approach, while potentially increasing the precision of the predictions, risks overlooking viable triples that fall marginally beyond this narrow margin, thereby potentially curtailing the comprehensive application of the voting strategy.

At an optimal interval of ± 0.15 , the model exhibited a superior F1 score of 68.56%, which underscores the efficacy of this interval size in accurately delineating uncertain predictions from their certain counterparts. This balance allows for a more nuanced application of the voting strategy, enhancing the model's ability to judiciously evaluate and incorporate ambiguous triples, thus optimizing overall performance.

Expanding the interval to ± 0.2 and further to ± 0.25 resulted in a gradual decline in F1 scores to 68.41% and 67.34%, respectively. This observation reveals a pivotal trade-off: While a broader interval incorporates a larger pool of uncertain predictions, it simultaneously increases the likelihood of misclassifying certain predictions as uncertain. Such an expansive approach may inadvertently inflate the volume of false positives, as the model begins to question predictions that would otherwise be

confidently accepted. This reduction in precision, aiming to encompass a wide range of uncertainties, highlights the essential need for a carefully calibrated interval. Such calibration must strike a balance between being inclusive of uncertain predictions and maintaining high accuracy in the results.

4.8.3 Hyperparameter Experiment 3: Data Augmentation Multiplier

We tested different values of the data augmentation multiplier, which determines the quantity of generated data, directly affecting the diversity and generalization capability of model training. The results are presented in [Table 10](#).

Table 10: Results of the hyperparameter experiment 3

Multiplier	P%	R%	F1%
0.5	68.69	67.0	67.83
0.75	68.62	67.89	68.25
1	69.31	67.83	68.56
1.25	68.07	67.87	67.97
1.5	66.84	66.15	66.49

When the multiplier was 0.5 and 0.75, the F1 scores were 67.83% and 68.25%, respectively. This indicates that moderate data augmentation can provide additional training samples, thereby contributing to an improvement in model performance.

At a multiplier of 1, the model achieved the optimal F1 score of 68.56%, showing that at this level of data augmentation, the model received enough training samples to enhance its generalization capability.

However, as the multiplier increased to 1.25 and 1.5, the F1 scores began to decline to 67.97% and 66.49%. Particularly at 1.5 times, the performance sharply decreased, which might be due to excessive data augmentation introducing more noise into the training data or the model overfitting to the generated data while neglecting the characteristics of the original dataset.

Regarding the potential issue of noise, it is a challenge that is difficult to completely avoid in the process of data augmentation. To minimize the occurrence of noise, we have implemented several strategies, including leveraging the strong correlation between attributes and sub-attributes, and designing specific prompts to restrict LLM from introducing new triples in the generated background text. Despite these measures, due to the high degree of freedom inherent in generative models, the emergence of some noise is inevitable. For instance, in the case where “the vehicle engine makes a noise when driving for a long time on rough terrain, leading to a transmission shaft malfunction,” our provided background was merely “when driving for a long time on rough terrain.” However, the LLM autonomously added “vehicle engine” and “noise,” thereby introducing a new, unannotated fault triple. Consequently, as the Data Augmentation Multiplier increases, the amount of noise correspondingly rises, which may diminish the accuracy of model predictions.

Regarding the aspect of overfitting leading to the neglect of subtle differences in the original dataset, let’s consider an example of a missed prediction in the text: “Observing this plug, it was found that two pins were inserted into the same socket,” which contains a triple (“two pins”, “part_failure”, “inserted into the same socket”). The failure entity “inserted into the same socket” does not belong to the common failure entities. Excessive augmentation may cause the model to generate a large volume

of data on common types of failures, such as mechanical failures, damage, wear and tear, etc., while overlooking these complex and instructive rare cases.

4.9 Case Analysis

In this section, we delve into specific examples to illustrate the effectiveness of our model.

For the relation semantic enhancement module, consider the triple (“car window”, “part_failure”, “occasionally auto-lowers”) which prior to the implementation of our innovative approach, the model struggled to accurately classify. By integrating the natural language description emphasizing “being faulty or damaged,” our model is now capable of understanding that the phenomenon of a “car window occasionally auto-lowering” signifies a “part_failure”. This nuanced comprehension allows the model to more effectively identify and categorize such anomalies as indicative of component malfunctions. This example underscores the enhanced precision our method brings to the identification of intricate relational contexts, thereby significantly improving the model’s ability to discern subtle nuances in data, a testament to the method’s efficacy in enhancing relational understanding.

Following the introduction of our second innovative component, the “voting strategy” aimed at reevaluating uncertain triples, we observed a marked improvement in the model’s ability to classify complex relations accurately. For instance, after applying the voting strategy, the triple (“soft, binary mattes”, “USED-FOR”, “partial scene segmentation”) from the sentence “We propose a novel step toward the unsupervised segmentation of whole objects by combining hints of partial scene segmentation offered by multiple soft, binary mattes.” shifted from a moderate to low confidence level to being correctly identified. This enhancement is attributed to the knowledge capability of LLMs, which can bolster understanding not just through context but also by drawing upon their extensive knowledge base. Consequently, “soft, binary mattes” became explicitly linked to the segmentation domain. This example highlights how our voting strategy, by harnessing the contextual and knowledge-based strengths of LLMs, significantly elevates the model’s proficiency in drawing nuanced connections, thereby enriching its interpretative depth and accuracy in identifying relation-specific nuances.

5 Conclusion

In this paper, we introduce a novel method for domain-specific triple extraction that utilizes relation semantic enhancement and fosters a tight collaboration between Large Language Models (LLMs) and Small Language Models (SLMs). Firstly, by employing relation semantic enhancement through a novel attention interaction module, our study deepens the understanding of semantic linkages between entities and relations in domain-specific relational triple extraction tasks. Secondly, our voting strategy innovatively harnesses LLMs to reevaluate predictions near confidence boundaries, initially identified by smaller models, through meticulously crafted prompts that evoke multiple-choice ‘votes’ for enhanced decision accuracy. Finally, data augmentation significantly contributes to relational triple extraction by generating rich and diverse domain-specific training samples, thereby expanding the model’s knowledge coverage and enhancing its generalization ability across complex relational contexts. Experiments on three domain-specific datasets show that our method outperforms baseline models in domain-specific relational triple extraction, demonstrating superior adaptability and effectiveness in tackling the issues of challenging samples and data scarcity. As future work, we aim to explore deeper collaboration between large and small models under domain-specific conditions, fully leveraging the powerful knowledge capabilities of LLMs, and seeking ways to minimize the noise issues arising from data augmentation.

Acknowledgement: I sincerely thank my professor, Jianpeng Hu.

Funding Statement: This research was funded by Science and Technology Innovation 2030—Major Project of “New Generation Artificial Intelligence” granted by Ministry of Science and Technology, Grant Number 2020AAA0109300.

Author Contributions: The authors confirm contribution to the paper as follows: Study conception and design: Jiakai Li; data collection: Jiakai Li, Geng Zhang; analysis and interpretation of results: Jiakai Li, Jianpeng Hu; draft manuscript preparation: Jiakai Li. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The code for this article is located at: <https://GitHub.com/lijk3023/ERTESD>. The data are all publicly available online. The CCL2022 dataset can be obtained from <https://GitHub.com/wgwang/CCL2022>. The SciERC dataset can be obtained from <https://nlp.cs.washington.edu/sciIE>. The CoNLL04 dataset can be obtained from <https://lavis.cs.hs-rm.de/storage/spert/public/datasets/conll04>.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] C. Peng, F. Xia, M. Naseriparsa, and F. Osborne, “Knowledge graphs: Opportunities and challenges,” *Artif. Intell. Rev.*, vol. 56, no. 11, pp. 13071–13102, Apr. 2023. doi: [10.1007/s10462-023-10465-9](https://doi.org/10.1007/s10462-023-10465-9).
- [2] A. Mahapatra, S. R. Nangi, A. Garimella, and N. Anandhavelu, “Entity extraction in low resource domains with selective pre-training of large language models,” in *Proc. EMNLP*, Abu Dhabi, United Arab Emirates, Dec. 2022, pp. 942–951.
- [3] X. Zhang *et al.*, “Domain-specific NER via retrieving correlated samples,” in *Proc. COLING*, Gyeongju, Korea, Oct. 2022, pp. 2398–2404.
- [4] D. Zelenko, C. Aone, and A. Richardella, “Kernel methods for relation extraction,” *J. Mach. Learn. Res.*, vol. 3, pp. 1083–1106, Feb. 2003.
- [5] Y. S. Chan and D. Roth, “Exploiting syntactico-semantic structures for relation extraction,” in *Proc. ACL*, Portland, OR, USA, Jun. 2011, pp. 551–560.
- [6] Q. Li and H. Ji, “Incremental joint extraction of entity mentions and relations,” in *Proc. ACL*, Baltimore, MD, USA, Jun. 2014, pp. 402–412.
- [7] S. Zheng, F. Wang, H. Bao, Y. Hao, P. Zhou and B. Xu, “Joint extraction of entities and relations based on a novel tagging scheme,” in *Proc. ACL*, Vancouver, Canada, Jul. 2017, pp. 1227–1236.
- [8] B. Xu *et al.*, “EmRel: Joint representation of entities and embedded relations for multi-triple extraction,” in *Proc. NAACL*, Seattle, WA, USA, Jul. 2022, pp. 659–665.
- [9] F. Ren, L. Zhang, X. Zhao, S. Yin, S. Liu and B. Li, “A simple but effective bidirectional framework for relational triple extraction,” in *Proc. WSDM '22*, Tempe, AZ, USA, Feb. 21–25, 2022, pp. 824–832.
- [10] J. Ning, Z. Yang, Y. Sun, Z. Wang, and H. Lin, “OD-RTE: A one-stage object detection framework for relational triple extraction,” in *Proc. ACL*, Toronto, Canada, Association for Computational Linguistics, Jul. 2023, pp. 11120–11135.
- [11] L. Yuan, Y. Cai, J. Wang, and Q. Li, “Joint multimodal entity-relation extraction based on edge-enhanced graph alignment network and word-pair relation tagging,” in *Proc. AAAI '23*, Washington DC, USA, Feb. 7–14, 2023, vol. 37, no. 9, pp. 11051–11059. doi: [10.1609/aaai.v37i9.26309](https://doi.org/10.1609/aaai.v37i9.26309).
- [12] J. Yan, J. Chen, Y. Wu, D. Z. Chen, and J. Wu, “T2G-FORMER: Organizing tabular features into relation graphs promotes heterogeneous feature interaction,” in *Proc. AAAI '23*, Washington DC, USA, Feb. 7–14 2023, vol. 37, no. 9, pp. 10720–10728. doi: [10.1609/aaai.v37i9.26272](https://doi.org/10.1609/aaai.v37i9.26272).

- [13] C. Qin, A. Zhang, Z. Zhang, J. Chen, M. Yasunaga and D. Yang, "Is ChatGPT a general-purpose natural language processing task solver?," in *Proc. EMNLP*, Singapore, Dec. 2023, pp. 1339–1384.
- [14] D. D. Palma, "Retrieval-augmented recommender system: Enhancing recommender systems with large language models," in *Proc. RecSys '23*, Singapore, Sep. 2023, pp. 1369–1373.
- [15] Y. Ma, Y. Cao, Y. Hong, and A. Sun, "Large language model is not a good few-shot information extractor, but a good reranker for hard samples!," in *Find. Assoc. Comput. Linguist.: EMNLP 2023*, Singapore, Dec. 2023, pp. 10572–10601.
- [16] P. Gupta, H. Schütze, and B. Andrassy, "Table filling multi-task recurrent neural network for joint entity and relation extraction," in *Proc. COLING*, Dec. 2016, pp. 2537–2547.
- [17] A. Katiyar and C. Cardie, "Going out on a limb: Joint extraction of entity mentions and relations without dependency trees," in *Proc. ACL*, Vancouver, Canada, Jul. 2017, pp. 917–928.
- [18] X. Zeng, D. Zeng, S. He, K. Liu, and J. Zhao, "Extracting relational facts by an end-to-end neural model with copy mechanism," in *Proc. ACL*, Melbourne, Australia, Jul. 2018, pp. 506–514.
- [19] Z. Wei, J. Su, Y. Wang, Y. Tian, and Y. Chang, "A novel cascade binary tagging framework for relational triple extraction," in *Proc. ACL*, Jul. 2020, pp. 1476–1488.
- [20] X. Li, X. Luo, C. Dong, D. Yang, B. Luan and Z. He, "TDEER: An efficient translating decoding schema for joint extraction of entities and relations," in *Proc. EMNLP*, Punta Cana, Dominican Republic, Nov. 2021, pp. 8055–8064.
- [21] Y. Wang, B. Yu, Y. Zhang, T. Liu, H. Zhu, and L. Sun, "TPLinker: Single-stage joint extraction of entities and relations through token pair linking," in *Proc. COLING*, Barcelona, Spain, Dec. 2020, pp. 1572–1582.
- [22] Z. Zhong and D. Chen, "A Frustratingly easy approach for entity and relation extraction," in *Proc. NAACL*, Jun. 2021, pp. 50–61.
- [23] M. Eberts and A. Ulges, "Span-based joint entity and relation extraction with transformer pre-training," in *ECAI 2020*, Santiago de Compostela, Spain, vol. 8, Aug. 29–Sep. 8, 2020, pp. 2006–2013.
- [24] H. Zheng *et al.*, "PRGC: Potential relation and global correspondence based joint relational triple extraction," in *Proc. ACL-IJCNLP*, Aug. 2021, pp. 6225–6235.
- [25] Y. Wang, C. Sun, Y. Wu, H. Zhou, L. Li and J. Yan, "UniRE: A unified label space for entity relation extraction," in *Proc. ACL-IJCNLP*, Aug. 2021, pp. 220–231.
- [26] X. Huang *et al.*, "Document-Level relation extraction via pair-aware and entity-enhanced representation learning," in *Proc. COLING*, Gyeongju, Korea, Oct. 2022, pp. 2418–2428.
- [27] Y. M. Shang, H. Huang, and X. Mao, "OneRel: Joint entity and relation extraction with one module in one step," in *Proc. AAAI-22*, Vancouver, BC, Canada, vol. 36, Feb. 22–Mar. 1, 2022, pp. 11285–11293.
- [28] W. Tang *et al.*, "Unified representation and interaction for joint relational triple extraction," in *Proc. EMNLP*, Abu Dhabi, United Arab Emirates, Dec. 2022, pp. 7087–7099.
- [29] P. Yang, X. Cong, Z. Sun, and X. Liu, "Enhanced language representation with label knowledge for span extraction," in *Proc. EMNLP*, Punta Cana, Dominican Republic, Nov. 2021, pp. 4623–4635.
- [30] T. B. Brown *et al.*, "Language models are few-shot learners," in *Proc. NIPS*, Vancouver, BC, Canada, Dec. 2020, pp. 1877–1901.
- [31] L. Ouyang *et al.*, "Training language models to follow instructions with human feedback," *Advances Neural Inform. Process. Syst.*, vol. 35, pp. 27730–27744, Mar. 2022.
- [32] R. Martínez-Cruz, A. J. López-López, and J. Portela, "ChatGPT vs state-of-the-art models: A benchmarking study in keyphrase generation task," arXiv preprint arXiv:2304.14177, Apr. 2023.
- [33] W. Sun *et al.*, "Is ChatGPT good at search? investigating large language models as re-ranking agents," in *Proc. EMNLP*, Singapore, Dec. 2023, pp. 14918–14937.
- [34] X. Wei *et al.*, "Zero-shot information extraction via chatting with ChatGPT," arXiv preprint arXiv:2302.10205, Feb. 2023.
- [35] R. Tang, X. Han, X. Jiang, and X. Hu, "Does synthetic data generation of LLMs help clinical text mining?" arXiv preprint arXiv:2303.04360, Apr. 2023.
- [36] C. H. Chiang and H. Y. Lee, "Can large language models be an alternative to human evaluations?" in *Proc. ACL*, Toronto, Canada, Jul. 2023, pp. 15607–15631.

- [37] S. Y. Feng *et al.*, “A survey of data augmentation approaches for NLP,” in *Findings Assoc. Comput. Linguist.: ACL-IJCNLP 2021*, Aug. 2021, pp. 968–988.
- [38] B. Ding *et al.*, “Is GPT-3 a good data annotator?,” in *Proc. ACL*, Toronto, Canada, Jul. 2023, pp. 11173–11195.
- [39] Y. Yu *et al.*, “Large language model as attributed training data generator: A tale of diversity and bias,” *Advances Neural Inform. Process. Syst.*, vol. 36, Oct. 2023. doi: [10.48550/arXiv.2306.15895](https://doi.org/10.48550/arXiv.2306.15895).
- [40] J. Chung, E. Kamar, and S. Amershi, “Increasing diversity while maintaining accuracy: Text data generation with large language models and human interventions,” in *Proc. ACL*, Toronto, Canada, Jul. 2023, pp. 575–593.
- [41] J. Devlin, M. W. Chang, K. Lee and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Proc. NAACL*, Minneapolis, Minnesota, USA, Jun. 2019, pp. 4171–4186.
- [42] W. Ye, B. Li, R. Xie, Z. Sheng, L. Chen and S. Zhang, “Exploiting entity BIO tag embeddings and multi-task learning for relation extraction with imbalanced data,” in *Proc. ACL*, Florence, Italy, Jul. 2019, pp. 1351–1360.
- [43] Y. Luan, L. He, M. Ostendorf, and H. Hajishirzi, “Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction,” in *Proc. EMNLP*, Brussels, Belgium, Oct.–Nov. 2018, pp. 3219–3232.
- [44] D. Roth and W. T. Yih, “A linear programming formulation for global inference in natural language tasks,” in *Proc. CoNLL 2004*, Boston, Massachusetts, USA, Association for Computational Linguistics, May 6–7, 2004, pp. 1–8.
- [45] D. Ye, Y. Lin, P. Li, and M. Sun, “Packed levitated marker for entity and relation extraction,” in *Proc. ACL*, Dublin, Ireland, May 2022, pp. 4904–4917.
- [46] Y. Shen, X. Ma, Y. Tang, and W. Lu, “A trigger-sense memory flow framework for joint entity and relation extraction,” in *Proc. WWW2021*, Ljubljana, Slovenia, Apr. 19–23, 2021, pp. 1704–1715.