A Survey of Generative Information Extraction

Zikang Zhang[1], Wangjie You[1], Tianci Wu[1], Xinrui Wang[1], Juntao Li[1]*, Min Zhang[1]
[1]Institute of Computer Science and Technology, Soochow University, China
{zkzhangzz,20234227017,20234227013,20234227007}@stu.suda.edu.cn; {ljt,minzhang}@suda.edu.cn

Abstract

Generative information extraction (Generative IE) aims to generate structured text sequences from unstructured text using a generative framework. Scaling in model size yields variations in adaptation and generalization, and also drives fundamental shifts in the techniques and approaches used within this domain. In this survey, we first review generative information extraction (IE) methods based on pre-trained language models (PLMs) and large language models (LLMs), focusing on their adaptation and generalization capabilities. We also discuss the connection between these methods and these two aspects. Furthermore, to balance task performance with the substantial computational demands associated with LLMs, we emphasize the importance of model collaboration. Finally, given the advanced capabilities of LLMs, we explore methods for integrating diverse IE tasks into unified models.

1 Introduction

Information Extraction (IE) (Wilks, 1997) is a popular and fundamental task in natural language processing, which aims to extract structured information from unstructured plain text. IE typically includes Named Entity Recognition (NER), Relation Extraction (RE), and Event Extraction (EE) (Xu et al., 2023b). Given the precise and structured nature of IE target, traditional IE methods have primarily relied on extractive architectures, where models like BERT (Devlin et al., 2018) and RoBERTa (Su et al., 2022b) pinpoint specific spans of text to extract relevant information. However, as the complexity of IE tasks grows, extractive approaches often require highly specialized designs to handle intricate tasks effectively. In contrast, generative IE, which regards the target of IE as the text sequence and the target tokens are generated in a sequential manner, can alleviate the above problems. This paradigm shift allows for the generation of different types of information, such as entities, relations, and events, in a coherent manner.

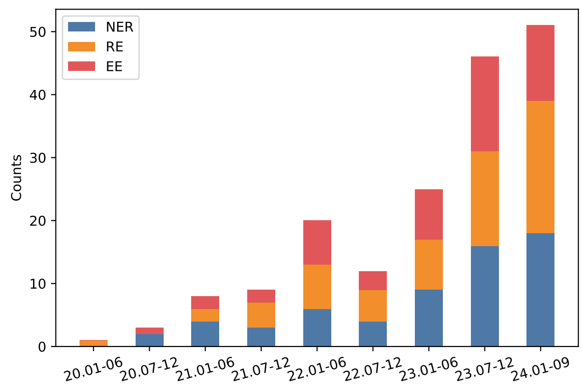


Figure 1: The number of papers on generative information extraction (IE) published from 2020 to the present (Referring to Appendix D for a detailed description).

The development of generative IE has been profoundly influenced by the principles outlined in scaling laws (Kaplan et al., 2020), which highlight that increasing model size leads to improved adaptability and generalization across a wide range of tasks (as shown in Figure 2). However, while scaling up models leads to better performance in IE tasks, it also incurs significant costs due to the increased computational demands of large parameter models (Tang et al., 2024; Zhao et al., 2023b). To balance performance with cost, it is essential to explore the collaboration between models with smaller and larger parameter counts. Finally, the integration of various IE tasks into a single model has become a prevailing trend, largely due to the robust capabilities of large language models (LLMs).

However, there is currently a lack of an in-depth review of existing PLMs-based generative IE methods. And existing surveys either focus solely on extractive IE, ignoring generative IE (Li et al., 2020; Wang et al., 2022b; Li et al., 2022b; Yang et al.,

*Corresponding author.

Proceedings of the 31st International Conference on Computational Linguistics, pages 4840–4870
January 19–24, 2025.

2022; Zhou et al., 2022); or they focus only on generative IE with LLMs, neglecting PLMs with small parameters (Xu et al., 2023b). In this survey, we provide a comprehensive review of existing generative IE methods based on PLMs. We mainly examine the widely utilized closed information extraction scenario, where the schema is predefined. The remainder of this survey is organized corresponding to the main steps in this procedure: (1) In Section 2: we introduced the Generative IE framework and discussed the core step of output linearization within this framework. We highlighted that model scaling is the key factor for the continuous improvement of IE performance. (2) In Section 3: We have summarized common methods for enhancing model adaptation and generalization capabilities in generative information extraction tasks. (3) In Section 4: We discussed collaboration for generative IE. (4) In Section 5: we examine methods of unified IE in the generative framework. (5) In Section 6: We conclude with related and future directions.

2 Generative Information Extraction

In this section, we will give a brief overview of generative IE. First, we introduce the formalization of the generative framework, Then, the instantiation of IE tasks within the generative framework and the benefits of generative IE will be introduced. Finally, we highlight several key factors in the generative framework: output linearization and model scaling. At the beginning, We define some terms. We denote the source sentence with n words as $X = (x_1, x_2, \ldots, x_n)$, the target sentence with m words as $Y = (y_1, y_2, \ldots, y_m)$, the generative model as $\mathcal{M}$, and the output target of IE with k elements in the source sentence as $\mathcal{T} = \{T_1, T_2, \ldots, T_k\}$.

2.1 Generative Framework

Task Formulation. Given a data point (X, Y), the objective of the generative framework is to learn a mapping function $f(\cdot)$ from the source sequence to the target sequence $f : X \to Y$ to estimate the unknown conditional distribution $P(Y|X;\theta)$, where θ denotes the parameter set of a model.

$$P(Y|X;\theta) = \prod_{i=1}^{m} P(y_i|y_{<i}, X;\theta) \quad (1)$$

where y_i is the token at the time step i and $y_{<i}$ are the tokens in previous $t-1$ decoding steps.

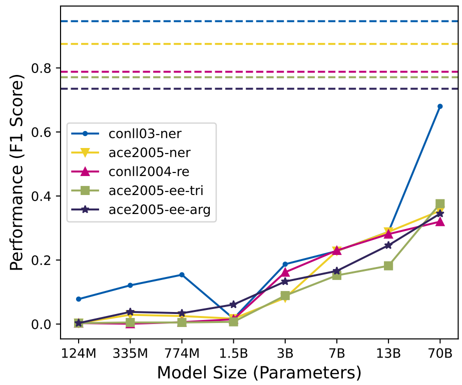


Figure 2: The performance of ICL on different datasets with different model sizes. The horizontal line represents the state-of-the-art (SOTA) of existing methods on this dataset. The blue and the yellow indicate the performance of the conll03 and ace2005 on the NER (conll03 SOTA: (Wang et al., 2020), ace2005 SOTA: (Yang et al., 2023)), the pink indicate the performance of the conll2004 on RE (conll04 SOTA: (Lou et al., 2023)), the green indicate the performance of the ace2005 on ED (ace2005 SOTA: (Wang et al., 2023b)), the purple indicate the performance of the ace2005 on EAE (ace2005 SOTA: (Hsu et al., 2021)). The experimental setup follows (Han et al., 2023b). The selected models are GPT2-base to GPT2-xl, openllama-3B-v2, and llama-2-7B, 13B, and 70B.

IE task instantiation. For NER, the T_i in $\mathcal{T}$ will be concretized as $T_i = (e_i, t_i)$, where e_i is a sub-sequence of X, t_i is an entity type (an example shown in Figure 5a). For RE, $T_i = (s_i, r_i, o_i)$, where s_i is the subject entity, o_i is the object entity, both of them are sub-sequence of X and r_i is relation type (an example shown in Figure 5b). A common subtask in RE is the Relation Classification (RC) task. And for EE, the T_i will be concretized as $T_i = (e_i, t_i, r_{i1}, a_{i1}, \ldots, r_{ij}, a_{ij})$, where e_i is event type, t_i is trigger words, r_{ij} is the jth argument role of event e_i and a_{ij} is the arguments of r_{ij} (an example shown in Figure 5c). Common subtasks in EE are Event Argument Extraction (EAE) and Event Dection (ED) tasks. Under the generative framework, the T_i will be converted into a text sequence, which can be generated by the generative model (Details in Appendix F). Therefore, various IE tasks can be unified naturally under the generative framework (Details in Appendix G, H).

The merits of generative IE. The advantages of the generative IE are as follows: (1) **General modeling and General task**: It is convenient to model various IE tasks (Fei et al., 2022). Specifically, when it faces complex IE structures, researchers will convert the output target into a text sequence (Hsu et al., 2021) without developing dedicated architectures. Furthermore, UIE (Universal information extraction) tasks can be naturally implemented in the generative framework (Wang et al., 2023b; Lu et al., 2022b). (2) **Knowledge sharing**: The above multi-task integration facilitates knowledge sharing between different IE tasks, enhancing the performance and the generalization ability of the model (Wang et al., 2022c; Lu et al., 2022b).

2.2 Key Factor of Generative IE

Output Linearization. Under the generative framework, the output space is not aligned between IE tasks and the generative model. The output space of IE tasks is the form of a set (Section 2.1), whereas that of generative models is in the form of natural language. Furthermore, the flexibility of natural language means that parsing out the output of the generative model to compare it to a gold target (to calculate standard metrics like precision, recall, and F1 score) is a non-trivial problem (Wadhwa et al., 2023). Typically, to alleviate the above issues, researchers convert the output target of IE tasks into structured text sequences (Details in Table 7), which are compatible with the output space of the generative model (Josifoski et al., 2021). We refer to the above process as **output linearization**. Meanwhile, the structured text sequences will be referred to as **linearized text**. Furthermore, output linearization can unify the task formats of different IE tasks. After obtaining the linearized text, researchers can design deterministic algorithms to extract the output targets of IE tasks from the linearized text (Athiwaratkun et al., 2020; Deußer et al., 2023; Ni et al., 2022; Cabot and Navigli, 2021; Josifoski et al., 2021). The format for linearized text in IE can be divided into natural language text sequence (Cui et al., 2021; Wang et al., 2022c), special token text sequence (Iovine et al., 2022; He and Tang, 2022), and code text sequence (Wang et al., 2022d; Li et al., 2023e) (Details in Appendix F). It is worth mentioning that code text sequences are typically used to align the output space of Code-LLMs (Li et al., 2023e; Sainz et al., 2023). Additionally, although special text sequences are in textual form, they are still "unnatural," resulting in a mismatch between the output format at pre-training time and inference time.

It is important to note that after output linearization, an order bias may be introduced, as structured objects in IE are concatenated into the target sequence in a pre-defined order. However, structured objects in IE constitute an unordered set (Zhang et al., 2022b; Xia et al., 2023; Li et al., 2023c).

Model Scaling. Since larger models often significantly enhance the adaptation and generalization capabilities of LMs (Brown et al., 2020; Wei et al., 2021), we investigate how scaling model size benefits generative IE. We evaluated the performance of models with varying parameter sizes in in-context learning (ICL) for different IE tasks, as detailed in Appendix K. Our findings indicate that once the parameter size exceeded a threshold of 1.5 billion, the ICL performance in IE tasks improved with further increases in parameter size. This suggests that the adaptability and generalization ability of the model in IE tasks enhance as the parameter size increases. However, as the number of model parameters increases, fine-tuning LLMs for IE tasks incurs significant computational overhead. Consequently, researchers have begun exploring methods to combine smaller parameter PLMs with LLMs to improve IE task performance while reducing costs. Additionally, due to the robust capabilities of LLMs, they are increasingly being used to handle multiple IE tasks simultaneously, making unified information extraction an emerging trend.

3 Adapation and Generalization

Adaptability and generalization have consistently been key focus areas in information extraction tasks (Details in Appendix A, B, C, I). Research indicates that PLMs have already demonstrated excellent adaptability and generalization in IE tasks. Furthermore, LLMs exhibit even stronger adaptability and generalization capabilities as the number of model parameters increases. In this section, we will explore various common methods for improving the adaptability and generalization capabilities of generative IE models, from PLMs to LLMs.

3.1 Training

A common strategy for enhancing the adaptability and generalization capabilities of information extraction (IE) models is fine-tuning. Various fine-tuning techniques influence the performance of the model in these areas. In the following sections, we will provide a detailed overview of existing fine-tuned generative IE methods and examine their impact on adaptability and generalization.

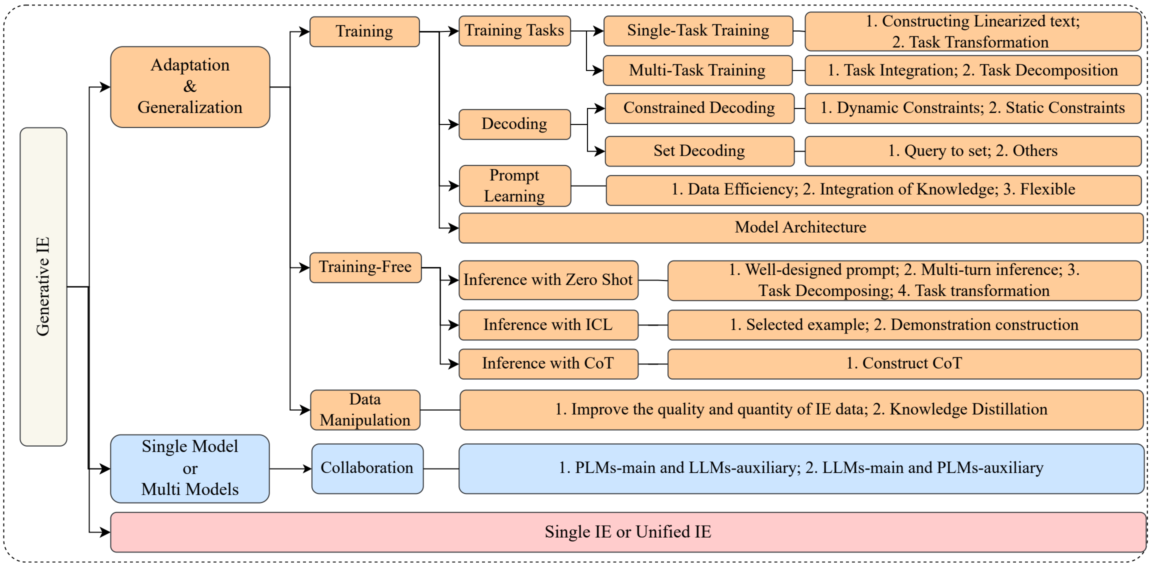


Figure 3: The framework of generative IE methods. The development trend of the generative IE methods as shown in Appendix E

3.1.1 Single-task and Multi-task training

The fine-tuning approach generally entails designing the input and output formats for the IE model and determining the appropriate number of training tasks. Variations in task formats and the number of tasks can enhance the adaptability and generalization of the model in IE tasks to different extents.

Single-task training. In generative information extraction tasks, single-task training typically involves either constructing a linearized text format for fine-tuning, which primarily enhances generalization or converting the information extraction task into a format suitable for fine-tuning, which improves both generalization and adaptability. **Construct Linearized Text.** Researchers develop various linearized text formats (as shown in the appendix Table 7) and fine-tune them directly to capture structural information. (Wadhwa et al., 2023; Ding et al., 2024a; Shi and Luo, 2024; Li et al., 2023f). Athiwaratkun et al. (2020), Ding et al. (2024a) and Paolini et al. (2021) finetune the generative model on the augmented natural language for NER. Nayak and Ng (2020), Cabot and Navigli (2021), Ni et al. (2022), Wadhwa et al. (2023), Shi and Luo (2024) and Tan et al. (2022) design a type of linearized text for RE, respectively. Compared to Nayak and Ng (2020), Cabot and Navigli (2021) considers the impact of the diversity of relation types. Tan et al. (2022) complete the missing relation triplets in the dataset. And Ni et al. (2022) explore various encoding representations for the source and target sequences for RC. Li et al. (2023a) enhance the semantics of the labels using paraphrase, inquiry, and synonym for RC. Hsu et al. (2021) construct a linearized text for each event type and fine-tune the generative model to generate linearized text for EE. **Task Transformation.** Reformulating the IE into other tasks not only helps to stimulate the capabilities of the model but also improves its adaptation and generalization ability. A common transformation method is converting IE tasks into QA tasks (Kondragunta et al., 2023; Kar et al., 2022; Uddin et al., 2024; Sainz et al., 2023), where questions are designed for each element type and answered to obtain the output target of IE. Furthermore, Wang et al. (2024a) consider multi-turn QA. Du and Ji (2022) retrieve the most similar QA pair to help answer questions. Unlike the above methods of manually constructing questions, Lu et al. (2023a) utilizes a question generation module to generate questions that can contain contextual information. Apart from converting IE to QA, Kim et al. (2022) transform RE to a template infilling task. A template is created for each relation type, using placeholders for the subject and object entities. Then, the generative model generates the content for the placeholders. Kan et al. (2023) decompose the IE task into subtasks, design a template that needs to be filled for each subtask, then concatenate the templates of all subtasks to form a prompt, and finally fill in this prompt. And Cui et al. (2021) reformulated NER as a template ranking problem. They use the generative model to score templates containing an entity type and a span. Further, the entity type contained in the highest-scoring template will be assigned to the span. Lu et al. (2022a) converted RC into a summarization formulation and use a summarization model to summarize the relation type in the source sentence. Then, they use a summarization model to summarize the relation type in the enhanced sentence, which enriches the semantics of the source sentence using the subject and object.

Multi-task training. Single-task training does not fully utilize the knowledge acquired during the pre-training phase. Moreover, knowledge can be shared among different IE tasks (Kan et al., 2023; Fei et al., 2022), and complex IE tasks can be decomposed into smaller, more manageable tasks (Gao et al., 2023; Duan et al., 2024; Zhou et al., 2023a). Consequently, researchers have adopted multi-task training, which facilitates knowledge sharing, enables rapid adaptation to new IE scenarios, and significantly enhances IE performance. **Task Integration.** Integrating different IE tasks or incorporating additional auxiliary tasks into a generative model facilitates knowledge sharing, thereby enhancing performance and generalization (Xiao et al., 2023; Li et al., 2024b). Lu et al. (2022b), Yu et al. (2023), Wang et al. (2022a), Lu et al. (2023b) and Wang et al. (2023b) unify different IE tasks into a generative model through output linearization. Cao and Ananiadou (2021) and Gan et al. (2023b) introduced the BIO tagging classification task and sentence classification task as auxiliary tasks, respectively. Iovine et al. (2022) and Chen et al. (2024c) proposed cyclic optimization by converting back and forth between the source sentence and the linearized text. And Li and Qian (2023) designed three generative meta-learning approaches to boost the generalization capability of generative models. Gan et al. (2023a) integrated text Classification, Sentiment Analysis and IE using a uniform input-output schema. **Task Decomposition.** Common decomposition methods for NER include entity recognition, entity type classification, or entity recognition based on a given entity type (Wang et al., 2022c; Gao et al., 2023). The RE task can usually be divided into entity recognition and relation classification (Gao et al., 2023; Lilong et al., 2024; Wu et al., 2024). And the EE task can be divided into ED and EAE (Lu et al., 2021; Zhou et al., 2023a; Duan et al., 2024). Wang et al. (2022c) design three subtasks related to NER, including generating all entity and entity type pairs, generating entities corresponding to a given entity type, and generating all entities. Gao et al. (2023) design a series of simple subtasks for each IE task to learn basic skills. Lu et al. (2021) adopt a curriculum learning approach, first learning the substructures of linearized text and then learning the complete linearized text for EE. Zhou et al. (2023a) decomposes the complex EE into multiple subtasks, i.e., extracting triggers and type, extracting arguments, and assigning arguments to corresponding roles. Duan et al. (2024) propose employing an auxiliary EKE sub-prompt and concurrently training both EE and EKE with the generative model.

3.1.2 Model architecture

The fine-tuning approach typically entails modifying the model architecture or incorporating additional modules. Designing intricate model architectures (He et al., 2023) for specific information extraction (IE) tasks can effectively enhance the capabilities of the model in those tasks. A common strategy involves introducing auxiliary modules to improve the adaptability and generalization of the model. For instance, Guo and Guo (2022) introduced the BERT-based Enhanced Lexicon Adapter to integrate external lexicon features into PLMs. Similarly, Fei et al. (2022) proposed a heterogeneous structure and an inductor structural broadcaster to fully leverage syntactic knowledge for UIE. Zhang et al. (2023e) introduced an entity start classification module to detect entity boundaries, while Shi et al. (2023) developed an event-type detector to pre-identify event types. Beyond auxiliary modules, Yang et al. (2021) designed a document-level encoder coupled with a multi-granularity decoder for document-level RE. Mo et al. (2023) devised a transformer architecture incorporating relation attention and type attention mechanisms.

3.1.3 Prompt learning

Prompt learning-based methods involve manually designing a prompt or inserting extra trainable modules into PLMs (Liu et al., 2023b; Li and Liang, 2021). Different from Section 3.1.2, prompt learning-based methods eliminate complex generative IE networks and massive extra parameters and allow the model to quickly transfer to different IE domains (Chen et al., 2023b) while achieving generalization capabilities similar to those of a fully parameterized model (Liu et al., 2023a).

An essential aspect of prompt learning methods is the process of constructing prompts, which encompasses both soft and hard prompts. A common approach in this process is the incorporation of knowledge, which can be accomplished through manual integration (Hsu et al., 2021; Su et al., 2022a; Song et al., 2023), the use of external tools (Song et al., 2023; Li et al., 2022a; Zhang et al., 2022a; Chen et al., 2023a; Cao et al., 2023), or the integration of knowledge from various domains or prompts (Chen et al., 2023b; Zhang et al., 2023b; Liu et al., 2022b; Wu et al., 2023). Hsu et al. (2021) design a template for each event type and learn to summarize the source sentence into a natural sentence following the predefined template. Song et al. (2023) construct a knowledge-enhanced soft prompt, which uses a relational graph neural network to encode event triplet entities and fuse them with word embedding to obtain a knowledge representation for RE. Chen et al. (2023b) fuse various source domain-prefix into a single prefix based on the similarity between the target domain and the source domain for the NER task. Zhang et al. (2023b) use prefix tuning to integrate overlap knowledge between different datasets and then learn special task knowledge through the adapter for EE. Liu et al. (2022b) and Wu et al. (2023) construct a context-and-type-aware prompt through attention mechanism. Nguyen et al. (2023) employs a graph attention mechanism to construct a context-and-aware prompt. Once a prompt is constructed, it typically allows for efficient data utilization, and is well-suited for low-resource scenarios (Duan et al., 2024). Furthermore, due to the flexibility of the prompt, it can easily facilitate knowledge transfer (Chen et al., 2021), thereby enhancing adaptability. Moreover, a well-designed prompt can effectively harness knowledge from PLMs, leading to improved performance (Chen et al., 2024d; Nguyen et al., 2023) and, consequently, better generalization (Chen et al., 2023b).

3.1.4 Decoding

Decoding is a critical aspect of generation. Beyond standard autoregressive decoding, alternative methods have been proposed that leverage the specific characteristics of IE, potentially enhancing the generalization capabilities of the model on IE tasks (Yan et al., 2021). In this section, we will discuss constrained decoding and set decoding (Figure 11).

Constrained Decoding. The output target of IE typically originates either from the source sentence or a predefined schema set. To prevent the generative model from producing tokens outside the intended scope, researchers have concentrated on imposing constraints on the model's generation process (Deußer et al., 2023; Lu et al., 2021; Cao and Ananiadou, 2021). The core principle of constrained decoding is to restrict the probability distribution associated with generating the ith token (Figure 11c). **Dynamic Constrained Decoding.** A common constrained method is to dynamically determine the distribution of the token in the current step based on a specific signal. Cao and Ananiadou (2021) utilized BIO tags as the signal. They first predict the BIO labels and then determine the distribution of the word list. They first predict the BIO tags corresponding to the token and then dynamically change the vocabulary distribution of the token based on the predicted BIO tags. Lu et al. (2021) and Deußer et al. (2023) used special tokens as the signal. When generating the ith token, Lu et al. (2021) proposes three optional vocabulary distributions for the token according to the special token: event schema, element string (event trigger words or arguments), and special token. Similarly, Deußer et al. (2023) determines whether to generate an entity type token, the end token, or any token from the source sentence based on the generated special token. Josifoski et al. (2021) employed the token generated in the previous steps as the signal. Specifically, they used a trie structure for constraint. And Liu et al. (2022a) utilized the action as the signal. **Static Constrained Decoding.** Another constrained method is the copy mechanism, which copies a token from a fixed scope. Yan et al. (2021) and Li et al. (2021b) employed the pointer network to complete copy mechanism. Zeng et al. (2018), Zeng et al. (2020), and Giorgi et al. (2022) also proposed similar work. Differently, Chang et al. (2023) mapped the hidden states of the generative model to a fixed scope through a linear layer.

Set Decoding. The output target of IE is essentially set where the elements are unordered (Section 2.1). However, the current generative IE methods force the element to be generated in a predefined order, which will suffer from error propagation, inefficient decoding, and order bias (Zhang et al., 2022b; Xia et al., 2023; Li et al., 2023c). Therefore researchers propose to directly generate the set using the generative framework. **Query to Set.** A common method for set generation is to use a query vector to generate an element in the set (as shown in Appendix Figure 11b) (Tan et al., 2021; Sui et al., 2023; Yang et al., 2021). Tan et al. (2021)

utilizes an entity query vector to predict entity and entity type. And Sui et al. (2023) also generates a relation triplet using a query vector as same as Tan et al. (2021). Furthermore, Chen et al. (2024b) proposed a dual-query approach. Ma et al. (2022) integrated the query into the template, generating all arguments corresponding to an event type at once. Besides utilizing query vector to achieve set generation, He and Tang (2022) treated each element as a target sequence and generated it in parallel. Li et al. (2023c) considered multiple permutations of output target of IE to optimize set probability approximately.

3.2 Training-Free

Due to the extensive volume of pre-training data, large language models (LLMs) already possess a rich knowledge base (Zhao et al., 2023b). This endows them with the potential for significant adaptability and generalization across various tasks. In this section, we will introduce methods to activate the adaptability and generalization capabilities of LLMs in IE tasks without additional training.

Inference with Zero Shot In the absence of sufficient data, emphasizing the construction of IE prompts, the simplification of IE tasks, and the use of multi-turn inference are crucial for optimizing the performance of LLMs and improving their adaptation and generalization. **Well-designed Prompt.** A well-constructed prompt, which generally encompasses the description of IE tasks, the format of the linearized text, and the incorporation of external IE knowledge, significantly aids in eliciting the information extraction capabilities of LLMs (Ni et al., 2023; Ashok and Lipton, 2023; Xie et al., 2023a). **Decomposing.** Decomposing IE tasks into simpler sub-tasks enables LLMs to more effectively address complex IE tasks. Xie et al. (2023a) decomposed NER by entity type, enabling LLMs to identify one type of entity at a time. Bian et al. (2023) first identified entities and then performed classification. In contrast, Wei et al. (2023) initially identified element types and subsequently identified the corresponding mentions based on these types. **Multi-turn Inference.** Leveraging LLMs to perform multi-turn inference is also a strategy to enhance their performance in IE. Ji (2023) and Wang et al. (2023a) pproposed a two-stage identification-correction framework for NER. Li et al. (2024a) introduced a three-step inference framework consisting of generation, clarification, and structuralization for generative IE. **Task Transformation.** In alignment with the discussion in Section 3.1.1, some studies reframe generative IE tasks into alternative task paradigms, such as Question Answering (QA) (Zhang et al., 2023a; Li et al., 2023b) and code generation (Guo et al., 2023; Bi et al., 2024; Li et al., 2023e; Wang et al., 2022d).

Inference with In Context Learning In data-scarce scenarios, researchers often employ in-context learning (ICL) to harness the capabilities of LLMs for completing IE tasks. **Demonstration Selection.** Research has demonstrated that selecting appropriate demonstration examples in ICL can significantly enhance task performance (Liu et al., 2021). Common methods for selecting high-quality data as demonstrations are typically based on the sentence or entity similarity (Rajpoot and Parikh, 2023; Wan et al., 2023; Wang et al., 2023a; Zhang et al., 2024b). In addition, Xie et al. (2023b) employed self-consistency (Wang et al., 2022e) to measure the quality of the data and select high-quality data. Mo et al. (2024) added negative examples in demonstrations. Qi et al. (2023) selected examples that can minimize the syntactic distribution difference between the test example and the LLMs as the demonstration. **Demonstration Format.** The demonstration format is also helpful in stimulating the capability of LLMs. Pang et al. (2023) included guidelines in examples to mitigate the issue of underspecified IE task descriptions.

Inference with Chain-of-Thought CoT (Wei et al., 2022) further incorporated step-by-step reasoning steps in each example to stimulate the reasoning potential of LLMs on IE. Ma et al. (2023b) manually constructed a CoT for RC. Zhao et al. (2023a) divided RE into multiple steps, determined the sequential relationship of each step and included the solution method for each step in a CoT.

3.3 Data Manipulation

In addition to designing fine-tuning methods, researchers also explore how to enhance the generalization and adaptability of the model in IE tasks from a data perspective. The quality and quantity of data are crucial for imparting knowledge to the model and improving the adaptation and generalization of the generative model. Moreover, some methods transfer the capabilities of a teacher model to a student model through data distillation.

Improve the quality and quantity of IE data. In PLMs, directly fine-tuning the model to generate data may lead to lower quality or reduced diversity in the generated outputs (Papanikolaou and Pierleoni, 2020). Therefore, there are many efforts proposed. Yaseen and Langer (2021) and Veyseh et al. (2023) employed back-translation and feedback mechanisms, respectively. Cabot and Navigli (2021) enriched the diversity of relation types in the dataset. Tan et al. (2022) further completed the missing relation triplets in the dataset. Hu et al. (2023b) designed two training tasks to maintain semantic and syntactic structure consistency. Song et al. (2024) and Guo et al. (2022) created augmented sentences from the corrupt sentences. Additionally, generating sentences from IE targets (i.e., reverse engineering) is also an effective method (Yili and Haonan (2023); Luo et al. (2024); Hu et al. (2022)). Hu et al. (2022) took an entity list as input and generates a sentence that includes all the entities from this list. Gui et al. (2024) constructed a schema-balance dataset, which includes positive schema, negative schema, and hard negative schema. In LLMs, a straightforward method is constructing a well-designed prompt for data generation (Chen et al., 2024a; Evuru et al., 2024; Meng et al., 2024; Xu et al., 2023c; Ye et al., 2024). However, due to the complexity of the IE output target, one-step data generation is not friendly for LLMs. Therefore, there are efforts to adopt the prompt pipeline approach for high-quality IE data generation (Gatto et al., 2024; Tang et al., 2023; Chen et al., 2023a; Cai et al., 2024; Sun et al., 2024; Luo et al., 2024). Similar to PLMs, reverse engineering is also used in LLMs for data generation (Ma et al., 2023a; Josifoski et al., 2023; Zhang et al., 2024a).

Knowledge Distillation. LLMs exhibit significant capabilities, however, employing these capabilities for generative IE is both costly and time-intensive (Zhou et al., 2023b). One approach to address this challenge is to distill the capabilities of LLMs into smaller models tailored for IE, which can be efficiently fine-tuned on few-shot training sets to enhance task-specific performance (Peng et al., 2024). A prevalent distillation method involves utilizing LLMs to generate or annotate datasets, thereby transferring the knowledge embedded in LLMs into the data, followed by training a meta-model on this dataset (Chen et al., 2024a; Bogdanov et al., 2024; Peng et al., 2024).

4 Single Model or Multi Models

As the number of model parameters increases, the performance of the model on IE tasks improves (Wang et al., 2022a; Ding et al., 2024a), significantly facilitating the completion of these tasks. However, utilizing large-parameter models (e.g., ChatGPT, GPT-4) requires substantial resources. Meanwhile, fine-tuned small parameter PLMs can achieve excellent performance on IE tasks (Peng et al., 2024; Yan et al., 2021; Hsu et al., 2021), and the cost of fine-tuning PLMs is manageable. To balance performance and cost, researchers often combine the inference capabilities of LLMs with the fine-tuning of small parameter PLMs. In this section, we will introduce how PLMs and LLMs collaborate to accomplish IE tasks.

4.1 PLMs-extractive and LLMs-auxiliary

In this part, PLMs primarily perform the extraction tasks, while LLMs provide corresponding assistance throughout the extraction process. **LLMs as Data Generators/Annotators.** A primary approach to utilizing LLMs as auxiliary tools is to consider them as data generators or annotators. As described in Section 3.3, LLMs transfer knowledge into the dataset, which is then transferred to PLMs through the dataset (Zaratiana et al., 2023; Bogdanov et al., 2024; Peng et al., 2024). Additionally, in situations of data scarcity, LLMs can be employed to generate supplementary data to mitigate the issue (Xu et al., 2023a; Zhou et al., 2024). **LLMs as Discriminators/Correctors.** Besides, LLMs can also serve as discriminators or correctors to judge the correctness of PLMs-generated results and make corrections. Kim et al. (2024) proposed using LLMs and self-consistency (Wang et al., 2022e) to verify and correct the results of PLMs. Zhang et al. (2024d) and Ma et al. (2023c) utilized LLMs to correct low-confidence results obtained and solve complex examples, respectively.

4.2 LLMs-extractive and PLMs-auxiliary

Conversely, in this part, LLMs perform the primary extraction tasks, and PLMs provide assistance to LLMs. The knowledge acquired by the PLMs is thus transferred to the LLM, with the expectation that the LLM will make more accurate predictions by integrating the task-specific knowledge of the PLM with its own domain expertise. Li et al. (2023d) regarded PLMs as scorers to retrieve the knowledge most similar to the source sentence. Zhang et al. (2024c) utilized the PLMs to calibrate the results generated by the LLMs. Tang et al. (2024), Ding et al. (2024b) and Jiang et al. (2024) regarded the PLMs as teachers, which the prediction result of the PLMs as a part of the prompt, to transfer task knowledge to LLMs. Additionally, Fan et al. (2024) constructed a new evaluation method by collaborating with PLMs and LLMs.

5 Single IE or Unified IE

Before the emergence of LLMs, researchers were already focusing on unifying IE tasks (Fei et al., 2022; Yu et al., 2023), but most efforts remained concentrated on individual IE tasks. IE exhibits significant diversity (Section 2.1), leading researchers to design task-specific methods for different IE tasks. These task-specific solutions bring some problems: 1. Obstructing knowledge sharing. 2. Developing dedicated architectures. 3. High cost and time-consuming (Lu et al., 2022b). To address the aforementioned challenges, there is a growing trend toward unifying the modeling of various IE tasks. Benefiting from the powerful capabilities of LLMs, they are capable of handling various IE tasks. Furthermore, owing to the intrinsic capacity of generative frameworks to integrate IE tasks (Lu et al., 2022b), there is a prevailing trend towards universal information extraction (Wang et al., 2023b). In this section, we will introduce the UIE method under the generative framework.

In a training-free scenario, the typical approaches to completing UIE involve designing specific prompts tailored to different IE tasks and subsequently utilizing LLMs for inference (Xie et al., 2023a; Guo et al., 2023; Bi et al., 2024). Building on this foundation, additional techniques such as task decomposition and task transformation may also be considered (Wei et al., 2023).

In scenarios requiring training, whether utilizing PLMs or LLMs, the prevailing approach to achieving UIE typically involves employing output linearization to standardize the outputs across different IE tasks (PLMs: (Paolini et al., 2021; Lu et al., 2022b; Yu et al., 2023); natural language LLMs: (Wang et al., 2023b); code-based LLMs: (Sainz et al., 2023; Li et al., 2024b)). Furthermore, some other work has been proposed. In PLMs, Fei et al. (2022) further proposed heterogeneous structure and inductor structural broadcaster on the aforementioned basis to fully unleash the power of syntactic knowledge for UIE. Kan et al. (2023) unified different IE tasks through template filling. In NL-LLMs, Xiao et al. (2023) involved multi-turn instruction-tuning for UIE, Lu et al. (2023b) designed various instructions for UIE, Gui et al. (2024) constructed a schema-balanced IE dataset, and Lee et al. (2024) considered task decomposition and parallel training for UIE. In code-LLMs, Sainz et al. (2023) finetuned Code-LLMs with annotation guidelines to improve the zero-shot performance. Li et al. (2024b) designed a two-stage fine-tuning algorithm that enables LLMs to better understand and follow the form of schemas.

6 Future direction

After introducing the existing generative IE methods based on PLMs and LLMs, we further propose some promising research directions in this section. We expect it to provide valuable insights and promote the development of generative IE.

Long Document IE. The mentioned methods perform well when dealing with short texts or sentences (Giorgi et al., 2022; Huang et al., 2021; Du et al., 2022; Lilong et al., 2024), but when processing long documents such as legal documents, the task becomes more challenging due to the complexity and diversity of the information contained. How to better model long texts and extract information from them will be a future research direction.

OIE. OpenIE refers to extracting structured information from unstructured text without any predefined schema (Zhou et al., 2022). It has always been a challenging task. For PLMs with small parameters, it is difficult to complete OpenIE due to insufficient abilities and knowledge (Kolluru et al., 2020). For LLMs, with their extensive knowledge base and strong understanding ability, they have promoted the development of openIE (Lu et al., 2023b), but openIE remains a challenging task.

Low resource IE. In practical scenarios, there is often a lack of data, leading to what is known as low-resource IE. Although LLMs exhibit excellent zero-shot and few-shot capabilities, they still fall short of optimal performance and cannot be directly applied in practice. Therefore, enhancing the IE capabilities of LLMs under low-resource conditions is a promising direction for future research.

Multimodal IE. Multimodal information extraction is an emerging field in natural language processing that focuses on extracting meaningful information from data that spans multiple modalities, such as text, images, audio, and video. Unlike traditional IE, which primarily deals with text, multimodal IE aims to integrate and analyze information from various sources to provide a more comprehensive understanding of the content.

Limitation

There are several limitations of this work. Firstly, IE generally includes extractive IE and generative IE, and the methods of extractive IE have occupied a large part of the entire development process of IE. However, this survey focuses solely on generative IE. To gain a comprehensive understanding of the methods in IE tasks, we encourage referencing other surveys on extractive IE (Li et al., 2020; Wang et al., 2022b; Li et al., 2022b; Yang et al., 2022; Zhou et al., 2022; Xu et al., 2023b). Moreover, the descriptions in this survey are mostly brief in order to provide a more comprehensive coverage within page limits. Instead of presenting the works in unstructured sequences, we primarily organize them into meaningful structured groups. Our aim is for this work to serve as a reference, where readers can delve into the corresponding works for more detailed information. Finally, due to personal limitations and understanding, our grasp of the future development trends in IE may not be comprehensive, hence there might be some deviations in the trends and future work mentioned in this paper.

Acknowledgments

We want to thank all the anonymous reviewers for their valuable comments. This work was supported by the National Science Foundation of China (NSFC No. 62206194 and 62276077), the Priority Academic Program Development of Jiangsu Higher Education Institutions, the Natural Science Foundation of Jiangsu Province, China (Grant No. BK20220488), and Young Elite Scientists Sponsorship Program by CAST (2023QNRC001).

References

Nasser Alshammari and Saad Alanazi. 2021. The impact of using different annotation schemes on named entity recognition. *Egyptian Informatics Journal*, 22(3):295–302.

Dhananjay Ashok and Zachary C Lipton. 2023. Promptner: Prompting for named entity recognition. *arXiv preprint arXiv:2305.15444*.

Ben Athiwaratkun, Cicero Nogueira dos Santos, Jason Krone, and Bing Xiang. 2020. Augmented natural language for generative sequence labeling. *arXiv preprint arXiv:2009.13272*.

Genady Beryozkin, Yoel Drori, Oren Gilon, Tzvika Hartman, and Idan Szpektor. 2019. A joint named-entity recognizer for heterogeneous tag-sets using a tag hierarchy. *arXiv preprint arXiv:1905.09135*.

Zhen Bi, Jing Chen, Yinuo Jiang, Feiyu Xiong, Wei Guo, Huajun Chen, and Ningyu Zhang. 2024. Codekgc: Code language model for generative knowledge graph construction. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 23(3):1–16.

Junyi Bian, Jiaxuan Zheng, Yuyi Zhang, and Shanfeng Zhu. 2023. Inspire the large language model by external knowledge on biomedical named entity recognition. *arXiv preprint arXiv:2309.12278*.

Sergei Bogdanov, Alexandre Constantin, Timothée Bernard, Benoit Crabbé, and Etienne Bernard. 2024. Nuner: Entity recognition encoder pre-training via llm-annotated data. *arXiv preprint arXiv:2402.15343*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Pere-Lluís Huguet Cabot and Roberto Navigli. 2021. Rebel: Relation extraction by end-to-end language generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2370–2381.

Zefan Cai, Po-Nien Kung, Ashima Suvarna, Mingyu Derek Ma, Hritik Bansal, Baobao Chang, P Jeffrey Brantingham, Wei Wang, and Nanyun Peng. 2024. Improving event definition following for zero-shot event detection. *arXiv preprint arXiv:2403.02586*.

Jiarun Cao and Sophia Ananiadou. 2021. Generativere: Incorporating a novel copy mechanism and pretrained model for joint entity and relation extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2119–2126.

Pengfei Cao, Zhuoran Jin, Yubo Chen, Kang Liu, and Jun Zhao. 2023. Zero-shot cross-lingual event argument extraction with language-oriented prefix-tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 12589–12597.

Yee Seng Chan and Dan Roth. 2011. Exploiting syntactico-semantic structures for relation extraction. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pages 551–560.

Hongyang Chang, Hongfei Xu, Josef van Genabith, Deyi Xiong, and Hongying Zan. 2023. Joinerbart: joint entity and relation extraction with constrained decoding, representation reuse and fusion. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Jianxun Chen, Peng Chen, and Xuxu Wu. 2023a. Generating chinese event extraction method based on chatgpt and prompt learning. *Applied Sciences*, 13(17):9500.

Ruirui Chen, Chengwei Qin, Weifeng Jiang, and Dongkyu Choi. 2024a. Is a large language model a good annotator for event extraction? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17772–17780.

Xiang Chen, Lei Li, Shumin Deng, Chuanqi Tan, Changliang Xu, Fei Huang, Luo Si, Huajun Chen, and Ningyu Zhang. 2021. Lightner: A lightweight tuning paradigm for low-resource ner via pluggable prompting. *arXiv preprint arXiv:2109.00720*.

Xiang Chen, Lei Li, Shuofei Qiao, Ningyu Zhang, Chuanqi Tan, Yong Jiang, Fei Huang, and Huajun Chen. 2023b. One model for all domains: collaborative domain-prefix tuning for cross-domain ner. *arXiv preprint arXiv:2301.10410*.

Xiang Chen, Lei Li, Yuqi Zhu, Shumin Deng, Chuanqi Tan, Fei Huang, Luo Si, Ningyu Zhang, and Huajun Chen. 2024b. Sequence labeling as non-autoregressive dual-query set generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 167–176.

Zhen Chen, Jingping Liu, Deqing Yang, Yanghua Xiao, Huimin Xu, Zongyu Wang, Rui Xie, and Yunsen Xian. 2024c. Exploiting duality in open information extraction with predicate prompt. *arXiv preprint arXiv:2401.11107*.

Zhenbin Chen, Zhixin Li, Yufei Zeng, Canlong Zhang, and Huifang Ma. 2024d. Gap: A novel generative context-aware prompt-tuning method for relation extraction. *Expert Systems with Applications*, page 123478.

Yew Ken Chia, Lidong Bing, Soujanya Poria, and Luo Si. 2022. Relationprompt: Leveraging prompts to generate synthetic data for zero-shot relation triplet extraction. *arXiv preprint arXiv:2203.09101*.

Leyang Cui, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang. 2021. Template-based named entity recognition using bart. *arXiv preprint arXiv:2106.01760*.

Tobias Deußer, Lars Hillebrand, Christian Bauckhage, and Rafet Sifa. 2023. Informed named entity recognition decoding for generative language models. *arXiv preprint arXiv:2308.07791*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Yuyang Ding, Juntao Li, Pinzheng Wang, Zecheng Tang, Bowen Yan, and Min Zhang. 2024a. Rethinking negative instances for generative named entity recognition. *arXiv preprint arXiv:2402.16602*.

Zepeng Ding, Wenhao Huang, Jiaqing Liang, Deqing Yang, and Yanghua Xiao. 2024b. Improving recall of large language models: A model collaboration approach for relational triple extraction. *arXiv preprint arXiv:2404.09593*.

Kalpit Dixit and Yaser Al-Onaizan. 2019. Span-level model for relation extraction. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 5308–5314.

Xinya Du and Heng Ji. 2022. Retrieval-augmented generative question answering for event argument extraction. *arXiv preprint arXiv:2211.07067*.

Xinya Du, Sha Li, and Heng Ji. 2022. Dynamic global memory for document-level argument extraction. *arXiv preprint arXiv:2209.08679*.

Junwen Duan, Xincheng Liao, Ying An, and Jianxin Wang. 2024. Keyee: Enhancing low-resource generative event extraction with auxiliary keyword sub-prompt. *Big Data Mining and Analytics*, 7(2):547–560.

Chandra Kiran Reddy Evuru, Sreyan Ghosh, Sonal Kumar, Utkarsh Tyagi, Dinesh Manocha, et al. 2024. Coda: Constrained generation based data augmentation for low-resource nlp. *arXiv preprint arXiv:2404.00415*.

Yuchen Fan, Yantao Liu, Zijun Yao, Jifan Yu, Lei Hou, and Juanzi Li. 2024. Evaluating generative language models in information extraction as subjective question correction. *arXiv preprint arXiv:2404.03532*.

Hao Fei, Shengqiong Wu, Jingye Li, Bobo Li, Fei Li, Libo Qin, Meishan Zhang, Min Zhang, and Tat-Seng Chua. 2022. Lasuie: Unifying information extraction with latent adaptive structure-aware generative language model. *Advances in Neural Information Processing Systems*, 35:15460–15475.

Chengguang Gan, Qinghao Zhang, and Tatsunori Mori. 2023a. Giellm: Japanese general information extraction large language model utilizing mutual reinforcement effect. *arXiv preprint arXiv:2311.06838*.

Chengguang Gan, Qinghao Zhang, and Tatsunori Mori. 2023b. Sentence-to-label generation framework for multi-task learning of japanese sentence classification and named entity recognition. In *International Conference on Applications of Natural Language to Information Systems*, pages 257–270. Springer.

Chang Gao, Wenxuan Zhang, Wai Lam, and Lidong Bing. 2023. Easy-to-hard learning for information extraction. *arXiv preprint arXiv:2305.09193*.

Joseph Gatto, Parker Seegmiller, Omar Sharif, and Sarah M Preum. 2024. Mad libs are all you need: Augmenting cross-domain document-level event argument data. *arXiv preprint arXiv:2403.03304*.

John Giorgi, Gary D Bader, and Bo Wang. 2022. A sequence-to-sequence approach for document-level relation extraction. *arXiv preprint arXiv:2204.01098*.

Honghao Gui, Hongbin Ye, Lin Yuan, Ningyu Zhang, Mengshu Sun, Lei Liang, and Huajun Chen. 2024. Iepile: Unearthing large-scale schema-based information extraction corpus. *arXiv preprint arXiv:2402.14710*.

Biyang Guo, Yeyun Gong, Yelong Shen, Songqiao Han, Hailiang Huang, Nan Duan, and Weizhu Chen. 2022. Genius: Sketch-based language model pre-training via extreme and selective masking for text generation and augmentation. *arXiv preprint arXiv:2211.10330*.

Qian Guo and Yi Guo. 2022. Lexicon enhanced chinese named entity recognition with pointer network. *Neural Computing and Applications*, 34(17):14535–14555.

Yucan Guo, Zixuan Li, Xiaolong Jin, Yantao Liu, Yutao Zeng, Wenxuan Liu, Xiang Li, Pan Yang, Long Bai, Jiafeng Guo, et al. 2023. Retrieval-augmented code generation for universal information extraction. *arXiv preprint arXiv:2311.02962*.

Dongchen Han, Zhaoqian Zheng, Hui Zhao, Shanshan Feng, and Haiting Pang. 2023a. Span-based single-stage joint entity-relation extraction model. *Plos one*, 18(2):e0281055.

Jiale Han, Shuai Zhao, Bo Cheng, Shengkun Ma, and Wei Lu. 2022. Generative prompt tuning for relation classification. *arXiv preprint arXiv:2210.12435*.

Ridong Han, Tao Peng, Chaohao Yang, Benyou Wang, Lu Liu, and Xiang Wan. 2023b. Is information extraction solved by chatgpt? an analysis of performance, evaluation criteria, robustness and errors. *arXiv preprint arXiv:2305.14450*.

Yuxin He, Jingyue Hu, and Buzhou Tang. 2023. Revisiting event argument extraction: Can eae models learn better when being aware of event co-occurrences? *arXiv preprint arXiv:2306.00502*.

Yuxin He and Buzhou Tang. 2022. Setgner: general named entity recognition as entity set generation. In *Proceedings of the 2022 conference on empirical methods in natural language processing*, pages 3074–3085.

I Hsu, Kuan-Hao Huang, Elizabeth Boschee, Scott Miller, Prem Natarajan, Kai-Wei Chang, Nanyun Peng, et al. 2021. Degree: A data-efficient generation-based event extraction model. *arXiv preprint arXiv:2108.12724*.

Ruijuan Hu, Haiyan Liu, and Huijuan Zhou. 2023a. Role knowledge prompting for document-level event argument extraction. *Applied Sciences*, 13(5):3041.

Xuming Hu, Yong Jiang, Aiwei Liu, Zhongqiang Huang, Pengjun Xie, Fei Huang, Lijie Wen, and Philip S Yu. 2022. Entity-to-text based data augmentation for various named entity recognition tasks. *arXiv preprint arXiv:2210.10343*.

Xuming Hu, Aiwei Liu, Zeqi Tan, Xin Zhang, Chenwei Zhang, Irwin King, and Philip S Yu. 2023b. Gda: Generative data augmentation techniques for relation extraction tasks. *arXiv preprint arXiv:2305.16663*.

Kuan-Hao Huang, I-Hung Hsu, Tanmay Parekh, Zhiyu Xie, Zixuan Zhang, Prem Natarajan, Kai-Wei Chang, Nanyun Peng, and Heng Ji. 2024. Textee: Benchmark, reevaluation, reflections, and future challenges in event extraction. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 12804–12825.

Kung-Hsiang Huang, Sam Tang, and Nanyun Peng. 2021. Document-level entity-based extraction as template generation. *arXiv preprint arXiv:2109.04901*.

Quzhe Huang, Yanxi Zhang, and Dongyan Zhao. 2023. From simple to complex: A progressive framework for document-level informative argument extraction. *arXiv preprint arXiv:2310.16358*.

Andrea Iovine, Anjie Fang, Besnik Fetahu, Oleg Rokhlenko, and Shervin Malmasi. 2022. Cyclener: an unsupervised training approach for named entity recognition. In *Proceedings of the ACM Web Conference 2022*, pages 2916–2924.

Bin Ji. 2023. Vicunaner: Zero/few-shot named entity recognition using vicuna. *arXiv preprint arXiv:2305.03253*.

Guochao Jiang, Ziqin Luo, Yuchen Shi, Dixuan Wang, Jiaqing Liang, and Deqing Yang. 2024. Toner: Type-oriented named entity recognition with generative language model. *arXiv preprint arXiv:2404.09145*.

Martin Josifoski, Nicola De Cao, Maxime Peyrard, Fabio Petroni, and Robert West. 2021. Genie: Generative information extraction. *arXiv preprint arXiv:2112.08340*.

Martin Josifoski, Marija Sakota, Maxime Peyrard, and Robert West. 2023. Exploiting asymmetry for synthetic training data generation: Synthie and the case of information extraction. *arXiv preprint arXiv:2303.04132*.

Mina Esmail Zadeh Nojoo Kambar, Armin Esmaeilzadeh, and Maryam Heidari. 2022. A survey on deep learning techniques for joint named entities and relation extraction. In *2022 IEEE World AI IoT Congress (AIIoT)*, pages 218–224. IEEE.

Zhigang Kan, Linhui Feng, Zhangyue Yin, Linbo Qiao, Xipeng Qiu, and Dongsheng Li. 2023. A composable generative framework based on prompt learning for various information extraction tasks. *IEEE Transactions on Big Data*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Debanjana Kar, Sudeshna Sarkar, and Pawan Goyal. 2022. Arggen: Prompting text generation models for document-level event-argument aggregation. In *Findings of the Association for Computational Linguistics: AACL-IJCNLP 2022*, pages 399–404.

Bosung Kim, Hayate Iso, Nikita Bhutani, Estevam Hruschka, Ndapa Nakashole, and Tom Mitchell. 2022. Zero-shot triplet extraction by template infilling. *arXiv preprint arXiv:2212.10708*.

Seoyeon Kim, Kwangwook Seo, Hyungjoo Chae, Jinyoung Yeo, and Dongha Lee. 2024. Verifiner: Verification-augmented ner via knowledge-grounded reasoning with large language models. *arXiv preprint arXiv:2402.18374*.

Keshav Kolluru, Samarth Aggarwal, Vipul Rathore, Mausam, and Soumen Chakrabarti. 2020. IMoJIE: Iterative memory-based joint open information extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5871–5886, Online. Association for Computational Linguistics.

Murali Kondragunta, Olatz Perez-de Viñaspre, and Maite Oronoz. 2023. Improving and simplifying template-based named entity recognition. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 79–86.

Minho Lee, Junghyun Min, Woochul Lee, and Yeonsoo Lee. 2024. Structured language generation model for robust structure prediction. *arXiv preprint arXiv:2402.08971*.

Bo Li, Dingyao Yu, Wei Ye, Jinglei Zhang, and Shikun Zhang. 2023a. Sequence generation with label augmentation for relation extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 13043–13050.

Fei Li, ZhiChao Lin, Meishan Zhang, and Donghong Ji. 2021a. A span-based model for joint overlapped and discontinuous named entity recognition. *arXiv preprint arXiv:2106.14373*.

Guozheng Li, Peng Wang, and Wenjun Ke. 2023b. Revisiting large language models as zero-shot relation extractors. *arXiv preprint arXiv:2310.05028*.

Haochen Li, Tong Mo, Hongcheng Fan, Jingkun Wang, Jiaxi Wang, Fuhao Zhang, and Weiping Li. 2022a. Kipt: knowledge-injected prompt tuning for event detection. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1943–1952.

Jiangnan Li, Yice Zhang, Bin Liang, Kam-Fai Wong, and Ruifeng Xu. 2023c. Set learning for generative information extraction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13043–13052.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2020. A survey on deep learning for named entity recognition. *IEEE transactions on knowledge and data engineering*, 34(1):50–70.

Mingchen Li, M Chen, Huixue Zhou, and Rui Zhang. 2023d. Petailor: Improving large language model by tailored chunk scorer in biomedical triple extraction. *arXiv preprint arXiv:2310.18463*.

Peng Li, Tianxiang Sun, Qiong Tang, Hang Yan, Yuanbin Wu, Xuanjing Huang, and Xipeng Qiu. 2023e. Codeie: Large code generation models are better few-shot information extractors. *arXiv preprint arXiv:2305.05711*.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 73–82.

Qian Li, Jianxin Li, Jiawei Sheng, Shiyao Cui, Jia Wu, Yiming Hei, Hao Peng, Shu Guo, Lihong Wang, Amin Beheshti, et al. 2022b. A survey on deep learning event extraction: Approaches and applications. *IEEE Transactions on Neural Networks and Learning Systems*.

Sha Li, Heng Ji, and Jiawei Han. 2021b. Document-level event argument extraction by conditional generation. *arXiv preprint arXiv:2104.05919*.

Wanli Li and Tieyun Qian. 2023. Generative meta-learning for zero-shot relation triplet extraction. *arXiv preprint arXiv:2305.01920*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Xue Li, Fina Polat, and Paul Groth. 2023f. Do instruction-tuned large language models help with relation extraction?

Yinghao Li, Rampi Ramprasad, and Chao Zhang. 2024a. A simple but effective approach to improve structured language model output for information extraction. *arXiv preprint arXiv:2402.13364*.

Zixuan Li, Yutao Zeng, Yuxin Zuo, Weicheng Ren, Wenxuan Liu, Miao Su, Yucan Guo, Yantao Liu, Xiang Li, Zhilei Hu, et al. 2024b. Knowcoder: Coding structured knowledge into llms for universal information extraction. *arXiv preprint arXiv:2403.07969*.

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

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 7999–8009.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2021. What makes good in-context examples for gpt-3? *arXiv preprint arXiv:2101.06804*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023a. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

Tianyu Liu, Yuchen Jiang, Nicholas Monath, Ryan Cotterell, and Mrinmaya Sachan. 2022a. Autoregressive structured prediction with language models. *arXiv preprint arXiv:2210.14698*.

Xiao Liu, Heyan Huang, Ge Shi, and Bo Wang. 2022b. Dynamic prefix-tuning for generative template-based event extraction. *arXiv preprint arXiv:2205.06166*.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2023b. Gpt understands, too. *AI Open*.

Jie Lou, Yaojie Lu, Dai Dai, Wei Jia, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2023. Universal information extraction as unified semantic matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 13318–13326.

Di Lu, Shihao Ran, Joel Tetreault, and Alejandro Jaimes. 2023a. Event extraction as question generation and answering. *arXiv preprint arXiv:2307.05567*.

Jinghui Lu, Ziwei Yang, Yanjie Wang, Xuejing Liu, and Can Huang. 2024. Padellm-ner: Parallel decoding in large language models for named entity recognition. *arXiv preprint arXiv:2402.04838*.

Keming Lu, I Hsu, Wenxuan Zhou, Mingyu Derek Ma, Muhao Chen, et al. 2022a. Summarization as indirect supervision for relation extraction. *arXiv preprint arXiv:2205.09837*.

Keming Lu, Xiaoman Pan, Kaiqiang Song, Hongming Zhang, Dong Yu, and Jianshu Chen. 2023b. Pivoine: Instruction tuning for open-world information extraction. *arXiv preprint arXiv:2305.14898*.

Yaojie Lu, Hongyu Lin, Jin Xu, Xianpei Han, Jialong Tang, Annan Li, Le Sun, Meng Liao, and Shaoyi Chen. 2021. Text2event: Controllable sequence-to-structure generation for end-to-end event extraction. *arXiv preprint arXiv:2106.09232*.

Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2022b. Unified structure generation for universal information extraction. *arXiv preprint arXiv:2203.12277*.

Zhizhao Luo, Youchen Wang, Wenjun Ke, Rui Qi, Yikai Guo, and Peng Wang. 2024. Boosting llms with ontology-aware prompt for ner data augmentation. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 12361–12365. IEEE.

Mingyu Derek Ma, Xiaoxuan Wang, Po-Nien Kung, P Jeffrey Brantingham, Nanyun Peng, and Wei Wang. 2023a. Star: Improving low-resource information extraction by structure-to-text data generation with large language models. In *NeurIPS 2023 Workshop on Synthetic Data Generation with Generative AI*.

Xilai Ma, Jing Li, and Min Zhang. 2023b. Chain of thought with explicit evidence reasoning for few-shot relation extraction. *arXiv preprint arXiv:2311.05922*.

Yubo Ma, Yixin Cao, YongChing Hong, and Aixin Sun. 2023c. Large language model is not a good few-shot information extractor, but a good reranker for hard samples! *arXiv preprint arXiv:2303.08559*.

Yubo Ma, Zehao Wang, Yixin Cao, Mukai Li, Meiqi Chen, Kun Wang, and Jing Shao. 2022. Prompt for extraction? paie: Prompting argument interaction for event argument extraction. *arXiv preprint arXiv:2202.12109*.

Zihao Meng, Tao Liu, Heng Zhang, Kai Feng, and Peng Zhao. 2024. Cean: Contrastive event aggregation network with llm-based augmentation for event extraction. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 321–333.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using lstms on sequences and tree structures. *arXiv preprint arXiv:1601.00770*.

Ying Mo, Hongyin Tang, Jiahao Liu, Qifan Wang, Zenglin Xu, Jingang Wang, Wei Wu, and Zhoujun Li. 2023. Multi-task transformer with relation-attention and type-attention for named entity recognition. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Ying Mo, Jian Yang, Jiahao Liu, Shun Zhang, Jingang Wang, and Zhoujun Li. 2024. C-icl: Contrastive in-context learning for information extraction. *arXiv preprint arXiv:2402.11254*.

Tapas Nayak and Hwee Tou Ng. 2020. Effective modeling of encoder-decoder architecture for joint entity and relation extraction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8528–8535.

Chien Nguyen, Hieu Man, and Thien Nguyen. 2023. Contextualized soft prompts for extraction of event arguments. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4352–4361.

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. Joint event extraction via recurrent neural networks. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 300–309.

Jian Ni, Gaetano Rossiello, Alfio Gliozzo, and Radu Florian. 2022. A generative model for relation extraction and classification. *arXiv preprint arXiv:2202.13229*.

Xuanfan Ni, Piji Li, and Huayang Li. 2023. Unified text structuralization with instruction-tuned language models. *arXiv preprint arXiv:2303.14956*.

Chaoxu Pang, Yixuan Cao, Qiang Ding, and Ping Luo. 2023. Guideline learning for in-context information extraction. *arXiv preprint arXiv:2310.05066*.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cicero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. *arXiv preprint arXiv:2101.05779*.

Yannis Papanikolaou and Andrea Pierleoni. 2020. Dare: Data augmented relation extraction with gpt-2. *arXiv preprint arXiv:2004.13845*.

Letian Peng, Zilong Wang, Feng Yao, Zihan Wang, and Jingbo Shang. 2024. Metaie: Distilling a meta model from llm for all kinds of information extraction tasks. *arXiv preprint arXiv:2404.00457*.

Ji Qi, Kaixuan Ji, Xiaozhi Wang, Jifan Yu, Kaisheng Zeng, Lei Hou, Juanzi Li, and Bin Xu. 2023. Mastering the task of open information extraction with large language models and consistent reasoning environment. *arXiv preprint arXiv:2310.10590*.

Pawan Kumar Rajpoot and Ankur Parikh. 2023. Gptfinre: In-context learning for financial relation extraction using large language models. *arXiv preprint arXiv:2306.17519*.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the thirteenth conference on computational natural language learning (CoNLL-2009)*, pages 147–155.

Yubing Ren, Yanan Cao, Ping Guo, Fang Fang, Wei Ma, and Zheng Lin. 2023. Retrieve-and-sample: Document-level event argument extraction via hybrid retrieval augmentation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 293–306.

Yubing Ren, Yanan Cao, Hao Li, Yingjie Li, Zixuan ZM Ma, Fang Fang, Ping Guo, and Wei Ma. 2024. Deie: Benchmarking document-level event information extraction with a large-scale chinese news dataset. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 4592–4604.

Oscar Sainz, Iker García-Ferrero, Rodrigo Agerri, Oier Lopez de Lacalle, German Rigau, and Eneko Agirre. 2023. Gollie: Annotation guidelines improve zero-shot information-extraction. *arXiv preprint arXiv:2310.03668*.

Yongliang Shen, Zeqi Tan, Shuhui Wu, Wenqi Zhang, Rongsheng Zhang, Yadong Xi, Weiming Lu, and Yueting Zhuang. 2023. Promptner: Prompt locating and typing for named entity recognition. *arXiv preprint arXiv:2305.17104*.

Ge Shi, Yunyue Su, Yongliang Ma, and Ming Zhou. 2023. A hybrid detection and generation framework with separate encoders for event extraction. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3163–3180.

Zhengpeng Shi and Haoran Luo. 2024. Cre-llm: A domain-specific chinese relation extraction framework with fine-tuned large language model. *arXiv preprint arXiv:2404.18085*.

Hetian Song, Qingmeng Zhu, Zhipeng Yu, Jian Liang, and Hao He. 2023. Generative event extraction via internal knowledge-enhanced prompt learning. In *International Conference on Artificial Neural Networks*, pages 90–102. Springer.

Sihan Song, Furao Shen, and Jian Zhao. 2024. Ropda: Robust prompt-based data augmentation for low-resource named entity recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19017–19025.

Jianlin Su, Ahmed Murtadha, Shengfeng Pan, Jing Hou, Jun Sun, Wanwei Huang, Bo Wen, and Yunfeng Liu. 2022a. Global pointer: Novel efficient span-based approach for named entity recognition. *arXiv preprint arXiv:2208.03054*.

Ming-Hsiang Su, Chin-Wei Lee, Chi-Lun Hsu, and Ruei-Cyuan Su. 2022b. Roberta-based traditional chinese medicine named entity recognition model. In *Proceedings of the 34th Conference on Computational Linguistics and Speech Processing (ROCLING 2022)*, pages 61–66.

Ananya Subburathinam, Di Lu, Heng Ji, Jonathan May, Shih-Fu Chang, Avirup Sil, and Clare Voss. 2019. Cross-lingual structure transfer for relation and event extraction. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (emnlp-ijcnlp)*, pages 313–325.

Dianbo Sui, Xiangrong Zeng, Yubo Chen, Kang Liu, and Jun Zhao. 2023. Joint entity and relation extraction with set prediction networks. *IEEE Transactions on Neural Networks and Learning Systems*.

Qi Sun, Kun Huang, Xiaocui Yang, Rong Tong, Kun Zhang, and Soujanya Poria. 2024. Consistency guided knowledge retrieval and denoising in llms for zero-shot document-level relation triplet extraction. *arXiv preprint arXiv:2401.13598*.

Qingyu Tan, Lu Xu, Lidong Bing, Hwee Tou Ng, and Sharifah Mahani Aljunied. 2022. Revisiting docred–addressing the false negative problem in relation extraction. *arXiv preprint arXiv:2205.12696*.

Zeqi Tan, Yongliang Shen, Shuai Zhang, Weiming Lu, and Yueting Zhuang. 2021. A sequence-to-set network for nested named entity recognition. *arXiv preprint arXiv:2105.08901*.

Ruixiang Tang, Xiaotian Han, Xiaoqian Jiang, and Xia Hu. 2023. Does synthetic data generation of llms help clinical text mining? *arXiv preprint arXiv:2303.04360*.

Xuemei Tang, Jun Wang, and Qi Su. 2024. Small language model is a good guide for large language model in chinese entity relation extraction. *arXiv preprint arXiv:2402.14373*.

Sentence Relation Triplets. A two-agent game for zero-shot relation triplet extraction.

Md Nayem Uddin, Enfa Rose George, Eduardo Blanco, and Steven Corman. 2024. Asking and answering questions to extract event-argument structures. *arXiv preprint arXiv:2404.16413*.

Shikhar Vashishth, Rishabh Joshi, Sai Suman Prayaga, Chiranjib Bhattacharyya, and Partha Talukdar. 2018. Reside: Improving distantly-supervised neural relation extraction using side information. *arXiv preprint arXiv:1812.04361*.

Amir Pouran Ben Veyseh, Franck Dernoncourt, Bonan Min, and Thien Nguyen. 2023. Generating labeled data for relation extraction: A meta learning approach with joint gpt-2 training. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11466–11478.

Somin Wadhwa, Silvio Amir, and Byron C Wallace. 2023. Revisiting relation extraction in the era of large language models. In *Proceedings of the conference. Association for Computational Linguistics. Meeting*, volume 2023, page 15566. NIH Public Access.

Zhen Wan, Fei Cheng, Zhuoyuan Mao, Qianying Liu, Haiyue Song, Jiwei Li, and Sadao Kurohashi. 2023. Gpt-re: In-context learning for relation extraction using large language models. *arXiv preprint arXiv:2305.02105*.

Chenguang Wang, Xiao Liu, Zui Chen, Haoyun Hong, Jie Tang, and Dawn Song. 2022a. Deepstruct: Pretraining of language models for structure prediction. *arXiv preprint arXiv:2205.10475*.

Hailin Wang, Ke Qin, Rufai Yusuf Zakari, Guoming Lu, and Jin Yin. 2022b. Deep neural network-based relation extraction: an overview. *Neural Computing and Applications*, pages 1–21.

Liwen Wang, Rumei Li, Yang Yan, Yuanmeng Yan, Sirui Wang, Wei Wu, and Weiran Xu. 2022c. Instructionner: A multi-task instruction-based generative framework for few-shot ner. *arXiv preprint arXiv:2203.03903*.

Lulu Wang, Kai Yu, Aishan Wumaier, Peng Zhang, Tuergen Yibulayin, Xi Wu, Jibing Gong, and Maihemuti Maimaiti. 2024a. Genre: generative multi-turn question answering with contrastive learning for entity–relation extraction. *Complex & Intelligent Systems*, pages 1–15.

Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. 2023a. Gpt-ner: Named entity recognition via large language models. *arXiv preprint arXiv:2304.10428*.

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

Xingyao Wang, Sha Li, and Heng Ji. 2022d. Code4struct: Code generation for few-shot event structure prediction. *arXiv preprint arXiv:2210.12810*.

Xinyu Wang, Yong Jiang, Nguyen Bach, Tao Wang, Zhongqiang Huang, Fei Huang, and Kewei Tu. 2020. Automated concatenation of embeddings for structured prediction. *arXiv preprint arXiv:2010.05006*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022e. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Yucheng Wang, Bowen Yu, Yilin Liu, and Shudong Lu. 2024b. True-uie: Two universal relations unify information extraction tasks. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1863–1876.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, et al. 2023. Zero-shot information extraction via chatting with chatgpt. *arXiv preprint arXiv:2302.10205*.

Zhepei Wei, Jianlin Su, Yue Wang, Yuan Tian, and Yi Chang. 2019. A novel cascade binary tagging framework for relational triple extraction. *arXiv preprint arXiv:1909.03227*.

Yorick Wilks. 1997. Information extraction as a core language technology. In *Information Extraction A Multidisciplinary Approach to an Emerging Information Technology: International Summer School, SCIE-97 Frascati, Italy, July 14–18, 1997*, pages 1–9. Springer.

Haolun Wu, Ye Yuan, Liana Mikaelyan, Alexander Meulemans, Xue Liu, James Hensman, and Bhaskar Mitra. 2024. Structured entity extraction using large language models. *arXiv preprint arXiv:2402.04437*.

Tongtong Wu, Fatemeh Shiri, Jingqi Kang, Guilin Qi, Gholamreza Haffari, and Yuan-Fang Li. 2023. Kc-gee: knowledge-based conditioning for generative event extraction. *World Wide Web*, 26(6):3983–3999.

Yu Xia, Yongwei Zhao, Wenhao Wu, and Sujian Li. 2023. Debiasing generative named entity recognition by calibrating sequence likelihood. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1137–1148.

Xinglin Xiao, Yijie Wang, Nan Xu, Yuqi Wang, Hanxuan Yang, Minzheng Wang, Yin Luo, Lei Wang, Wenji Mao, and Daniel Zeng. 2023. Yayi-uie: A chat-enhanced instruction tuning framework for universal information extraction. *arXiv preprint arXiv:2312.15548*.

Tingyu Xie, Qi Li, Jian Zhang, Yan Zhang, Zuozhu Liu, and Hongwei Wang. 2023a. Empirical study of zero-shot ner with chatgpt. *arXiv preprint arXiv:2310.10035*.

Tingyu Xie, Qi Li, Yan Zhang, Zuozhu Liu, and Hongwei Wang. 2023b. Self-improving for zero-shot named entity recognition with large language models. *arXiv preprint arXiv:2311.08921*.

Benfeng Xu, Quan Wang, Yajuan Lyu, Dai Dai, Yongdong Zhang, and Zhendong Mao. 2023a. S2ynre: Two-stage self-training with synthetic data for low-resource relation extraction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8186–8207.

Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, and Enhong Chen. 2023b. Large language models for generative information extraction: A survey. *arXiv preprint arXiv:2312.17617*.

Xin Xu, Yuqi Zhu, Xiaohan Wang, and Ningyu Zhang. 2023c. How to unleash the power of large language models for few-shot relation extraction? *arXiv preprint arXiv:2305.01555*.

Hang Yan, Tao Gui, Junqi Dai, Qipeng Guo, Zheng Zhang, and Xipeng Qiu. 2021. A unified generative framework for various ner subtasks. *arXiv preprint arXiv:2106.01223*.

Hang Yang, Dianbo Sui, Yubo Chen, Kang Liu, Jun Zhao, and Taifeng Wang. 2021. Document-level event extraction via parallel prediction networks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6298–6308.

Yang Yang, Zhilei Wu, Yuexiang Yang, Shuangshuang Lian, Fengjie Guo, and Zhiwei Wang. 2022. A survey of information extraction based on deep learning. *Applied Sciences*, 12(19):9691.

Yawen Yang, Xuming Hu, Fukun Ma, Aiwei Liu, Lijie Wen, S Yu Philip, et al. 2023. Gaussian prior reinforcement learning for nested named entity recognition. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Usama Yaseen and Stefan Langer. 2021. Data augmentation for low-resource named entity recognition using backtranslation. *arXiv preprint arXiv:2108.11703*.

Junjie Ye, Nuo Xu, Yikun Wang, Jie Zhou, Qi Zhang, Tao Gui, and Xuanjing Huang. 2024. Llm-da: Data augmentation via large language models for few-shot named entity recognition. *arXiv preprint arXiv:2402.14568*.

Qian Yili and Xu Haonan. 2023. Datg: Data augmentation with transformer-based generation for low-resource named entity recognition. In *2023 China Automation Congress (CAC)*, pages 6188–6193. IEEE.

Xin Cong Yu, Mengcheng Fang, Tingwen Liu, Haiyang Yu, Zhongkai Hu, Fei Huang, Yongbin Li, Bin Wang, et al. 2023. Universal information extraction with meta-pretrained self-retrieval. *arXiv preprint arXiv:2306.10444*.

Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. 2023. Gliner: Generalist model for named entity recognition using bidirectional transformer. *arXiv preprint arXiv:2311.08526*.

Daojian Zeng, Haoran Zhang, and Qianying Liu. 2020. Copymtl: Copy mechanism for joint extraction of entities and relations with multi-task learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 9507–9514.

Xiangrong Zeng, Daojian Zeng, Shizhu He, Kang Liu, and Jun Zhao. 2018. Extracting relational facts by an end-to-end neural model with copy mechanism. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 506–514.

Kai Zhang, Bernal Jiménez Gutiérrez, and Yu Su. 2023a. Aligning instruction tasks unlocks large language models as zero-shot relation extractors. *arXiv preprint arXiv:2305.11159*.

Kaihang Zhang, Kai Shuang, Xinyue Yang, Xuyang Yao, and Jinyu Guo. 2023b. What is overlap knowledge in event argument extraction? ape: A cross-datasets transfer learning model for eae. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 393–409.

Liqi Zhang, Lianhe Yang, Yinhao Bai, Hongfei Zhu, and Xingyu Liu. 2023c. R-tes: Regularized template style for generative joint relational triple extraction. In *Proceedings of the 2023 6th International Conference on Machine Learning and Natural Language Processing*, pages 21–26.

Manman Zhang, Shuocan Zhu, Jingmin Zhang, Yu Han, Xiaoxuan Zhu, and Leilei Zhang. 2024a. Entity relation joint extraction with data augmentation based on large language model. In *International Conference on Intelligent Information Processing*, pages 207–214. Springer.

Meishan Zhang, Bin Wang, Hao Fei, and Min Zhang. 2024b. In-context learning for few-shot nested named entity recognition. *arXiv preprint arXiv:2402.01182*.

Qing Zhang, Yuechen Yang, Hayilang Zhang, Zhengxin Gao, Hao Wang, Jianyong Duan, Li He, and Jie Liu. 2023d. Zero-shot relation triplet extraction via retrieval-augmented synthetic data generation. In *International Conference on Neural Information Processing*, pages 367–379. Springer.

Sheng Zhang, Patrick Ng, Zhiguo Wang, and Bing Xiang. 2022a. Reknow: enhanced knowledge for joint entity and relation extraction. *arXiv preprint arXiv:2206.05123*.

Shuai Zhang, Yongliang Shen, Zeqi Tan, Yiquan Wu, and Weiming Lu. 2022b. De-bias for generative extraction in unified ner task. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 808–818.

Xinliang Frederick Zhang, Carter Blum, Temma Choji, Shalin Shah, and Alakananda Vempala. 2024c. Ultra: Unleash llms' potential for event argument extraction through hierarchical modeling and pair-wise refinement. *arXiv preprint arXiv:2401.13218*.

Xinpeng Zhang, Ming Tan, Jingfan Zhang, and Wei Zhu. 2023e. Nag-ner: a unified non-autoregressive generation framework for various ner tasks. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 676–686.

Zhen Zhang, Yuhua Zhao, Hang Gao, and Mengting Hu. 2024d. Linkner: Linking local named entity recognition models to large language models using uncertainty. *arXiv preprint arXiv:2402.10573*.

Zhengkuan Zhang, Weiran Xu, and Qianqian Chen. 2016. Joint event extraction based on skip-window convolutional neural networks. In *Natural Language Understanding and Intelligent Applications: 5th CCF Conference on Natural Language Processing and Chinese Computing, NLPCC 2016, and 24th International Conference on Computer Processing of Oriental Languages, ICCPOL 2016, Kunming, China, December 2–6, 2016, Proceedings 24*, pages 324–334. Springer.

Hangtian Zhao, Hakiz Yilahun, and Askar Hamdulla. 2023a. Pipeline chain-of-thought: A prompt method for large language model relation extraction. In *2023 International Conference on Asian Language Processing (IALP)*, pages 31–36. IEEE.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023b. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Shaowen Zhou, Bowen Yu, Aixin Sun, Cheng Long, Jingyang Li, Haiyang Yu, Jian Sun, and Yongbin Li. 2022. A survey on neural open information extraction: Current status and future directions. *arXiv preprint arXiv:2205.11725*.

Sitong Zhou, Meliha Yetisgen, and Mari Ostendorf. 2023a. Building blocks for complex tasks: Robust generative event extraction for radiology reports under domain shifts. *arXiv preprint arXiv:2306.09544*.

Sizhe Zhou, Yu Meng, Bowen Jin, and Jiawei Han. 2024. Grasping the essentials: Tailoring large language models for zero-shot relation extraction. *arXiv preprint arXiv:2402.11142*.

Wenxuan Zhou, Sheng Zhang, Yu Gu, Muhao Chen, and Hoifung Poon. 2023b. Universalner: Targeted distillation from large language models for open named entity recognition. *arXiv preprint arXiv:2308.03279*.

A Experiment setting of adaptation and generalization in the IE task

Adaptability in information extraction tasks refers to the ability of the model to effectively extract the required information when confronted with data from different domains, often involving cross-domain adaptation. Generalization refers to the ability of the model to perform well on unseen data, leveraging the knowledge acquired during training. In information extraction tasks, experimental settings for assessing model adaptability typically include the following: Cross-Domain Adaptation, Zero-Shot Learning, Few-Shot Learning and Transfer Learning. And experimental settings for assessing model generalization ability typically include the following: Cross-Validation, Train-Test Split and Cross-Domain Evaluation.

B Statistics of datasets and benchmarks

Table 1 presents the commonly used datasets and benchmarks in IE tasks. The "Name of Benchmark" column lists the names of the benchmarks or datasets. The "IE Tasks" column identifies the information extraction tasks associated with each benchmark or dataset. The "Domain" column specifies the domain knowledge encompassed by the benchmark or dataset. In the "Number of Datasets" column, "multi" indicates that the benchmark comprises multiple datasets, while "single" indicates that it consists of only one dataset. The "Leaderboard" column uses a value of 0 to indicate that the benchmark lacks a leaderboard, and a value of 1 to indicate that a leaderboard is present. Finally, the "Train/Test" column provides the counts for the training and test sets included in the benchmark. The data for the leaderboard comes from the website: https://paperswithcode.com/

C Performance

We have evaluated the performance of generative IE methods based on PLMs and LLMs across various IE tasks, as illustrated in Table 2, 3, 4, 5, 6. The "Methods" column lists the generative IE methods, while the "Datasets" column specifies the relevant datasets. In the "Training" column, a value of 0 indicates that the method does not require training, whereas a value of 1 indicates that training is required. The "Experimental Setup" column uses "full" to denote training under full resource conditions, "low" to indicate training under low-resource conditions, "zero-shot" for LLM inference without any training samples, and "few-shot" for LLM inference with a limited number of samples. The "Model" column identifies the type of model used by each method, and the "Categories" column classifies the methods into relevant categories.

D Statistics on the number of related papers

To conduct a comprehensive survey of generative IE, we initiated our study by searching Google Scholar for relevant papers. We utilized the following keywords in our search: "named entity recognition" AND "generative," "relation extraction" AND "generative," and "event extraction" AND "generative." The search was confined to papers published between 2020 and 2024, and the results were further narrowed down to the top 50 entries for each keyword. For papers published in 2023 and 2024, we expanded our search criteria to include the keywords "named entity recognition" AND "LLMs," "relation extraction" AND "LLMs," and "event extraction" AND "LLMs." After compiling the search results, we manually filtered out irrelevant papers. The distribution of generative IE-related publications on Google Scholar over the past five years is illustrated in Figure 1. The observed trend is significant: with the advent of large language models, the number of papers on generative IE has been steadily increasing, underscoring the necessity of a comprehensive survey to review recent advancements in generative IE technology.

E The development of generative IE models

In Figure 4, the important and popular works along the generative IE development are shown in the timeline. We observe that most generative IE methods are concentrated in the Training Task category. Following the advent of LLMs, the number of methods in the Model Architecture and Prompt Learning categories has gradually declined, while new methods have emerged in the Inference with LLMs and Collaboration categories.

F Examples of Linearized text

To gain an intuitive understanding of linearized text in different forms (e.g., natural language text sequence, special token text sequence, and code text sequence), we have listed some examples for each form of linearized text in Table 7. Meanwhile, in

The Name of Benchmark	IE tasks	Domain	The number of Dataset	leaderboard	Train/Test
UniversalNER (Zhou et al., 2023b)	NER	General	Multi	0	-
TextEE (Huang et al., 2024)	EE	General	Multi	0	-
DEIE (Ren et al., 2024)	EE	General	Multi	0	-
InstructUIE (Wang et al., 2023b)	UIE	General	Multi	0	-
KnowCoder benchmark (Li et al., 2024b)	UIE	General	Multi	0	-
TRUE-UIE (Wang et al., 2024b)	UIE	General	Multi	0	-
IEPILE (Gui et al., 2024)	UIE	General	Multi	0	-
YAYI-UIE (Xiao et al., 2023)	UIE	General	Multi	0	-
CoNLL2003	NER	General	Single	1	14041/3453
OntoNotes	NER	General	Single	1	59924/8262
FewNERD	NER	General	Single	1	131767/37648
ACE2004	NER	General	Single	1	6202/812
ACE2005	NER	General	Single	1	7299/1060
Genia	NER	Biomed	Single	1	15023/1854
CADEC	NER	Biomed	Single	0	-
TACRED	RE	General	Single	1	68,124/15,509
SciERC	RE	Scientific	Single	1	2136/551
DocRED	RE	General	Single	1	3052/100
RE-DocRED	RE	General	Single	1	3053/500
FewRel	RE	Medical	Single	1	56k/14k
Wiki-ZSL	RE	General	Single	1	-
NYT	RE	General	Single	1	5.6k/5k
WebNLG	RE	General	Single	1	5019/703
TACRED-Revisit	RE	General	Single	1	58,465/13,418
CONLL04	RE	General	Single	1	922/288
ADE	RE	Biomed	Single	1	3417/428
RAMS	EE	General	Single	0	7,329/7,329
WIKIEVENTS	EE	General	Single	1	5,262/492
ACE05-E	EE	General	Single	0	17172/832
ACE05-E+	EE	General	Single	0	19216/676
ERE-EN	EE	General	Single	0	14736/ 1163
DocEE	EE	General	Single	0	22k/2.7k

Table 1: Statistics of datasets and benchmarks

Methods	Datasets			Train-ing?	Experimental Setup	Model	Category
	Conll03	Genia	CADEC				
BARTNER (Yan et al., 2021)	93.24	79.23	70.64	training	full	BART	Decoding
Metaretriever (Yu et al., 2023)	92.38	-	-	1	full	T5	Training Task
TANL (Paolini et al., 2021)	91.7	76.4	-	1	full	T5	Training Task
LasUIE (Fei et al., 2022)	93.2	-	-	1	full	T5	Training Task/Model Architecture
IE-E2H (Gao et al., 2023)	92.43	-	-	1	full	T5	Training Task
ASP (Liu et al., 2022a)	94.1	-	-	1	full	T5	Decoding
Universal-IE (Lu et al., 2022b)	92.99	-	-	1	full	T5	Training Task
DeepStruct (Wang et al., 2022a)	93.0	-	-	1	full	GLM10B	Training Task
InstructUIE (Wang et al., 2023b)	92.94	74.71	-	1	full	FlanT5-11B	Training Task
PaDeLLM (Lu et al., 2024)	92.52	85.02	-	1	full	llama2-7B/Baichuan-7B	-
GNER (Ding et al., 2024a)	93.60	-	-	1	full	llama2-7B	Training Task
UniversalNER (Zhou et al., 2023b)	93.30	77.54	-	1	full	llama2-7B	Data Manipulation
YAYI-UIE (Xiao et al., 2023)	96.77	75.21	-	1	full	Baichuan2-13B	Training Task
GOLLIE (Sainz et al., 2023)	93.1	-	-	1	full	Code-LLaMA-34B	Training Task
KnowCoder (Li et al., 2024b)	95.1	76.7	-	1	full	LLaMA2-base-7B	Training Task
ANL (Athiwaratkun et al., 2020)	91.48	-	-	1	full	T5	Training Task
De-Bias (Zhang et al., 2022b)	93.14	79.08	71.60	1	full	T5	Data Manipulation
Debiasing (Xia et al., 2023)	93.48	79.49	71.66	1	full	BART	-
InformedNER (Deußer et al., 2023)	91.51	-	-	1	full	GPT2	Decoding
LightNER (Chen et al., 2021)	92.93	-	-	1	full	BART	Prompt Learning
Multi-task transformer (Mo et al., 2023)	93.44	79.77	71.96	1	full	BART	Model Architecture
NAG-NER (Zhang et al., 2023e)	92.8	-	71.3	1	full	BANG	Model Architecture
SetGNER (He and Tang, 2022)	93.2	-	73.56	1	full	BART	Decoding
templateNER (Cui et al., 2021)	92.55	-	-	1	full	BART	Training Task
CODEIE (Li et al., 2023e)	82.32	-	-	0	few-shot(5-shot)	code-davinci-002	Inference with LLMs
C-ICL (Mo et al., 2024)	87.36	-	-	0	zero-shot	CodeLlama-34B	Inference with LLMs
GPT-NER (Wang et al., 2023a)	90.91	64.42	-	0	zero-shot	GPT3	Inference with LLMs
ChatGPT-IE (Han et al., 2023b)	60.10	38.09	-	0	zero-shot	chatgpt	Inference with LLMs
Self-Improving-for-NER (Xie et al., 2023b)	74.99	-	51.11	0	zero-shot	GPT-3.5	Inference with LLMs
VerifiNER (Kim et al., 2024)	-	55.46	-	0	few-shot	GPT3	Inference with LLMs
	MIT Moive	MIT Restaurant	ATIS				
templateNER (Cui et al., 2021)	52.2	58.7	92.6	1	low(50-shot)	BART	Training Task
CollaborativeNER (Chen et al., 2023b)	81.57	75.92	-	1	low(50-shot)	T5	Prompt Learning
InstructionNER (Wang et al., 2022c)	75.6	71.8	95.4	1	low(50-shot)	T5	Training Task
LightNER (Chen et al., 2021)	73.1	62.0	92.8	1	low(50-shot)	BART	Prompt Learning
InstructUIE (Wang et al., 2023b)	63.00	20.99	-	0	zero-shot	FlanT5 11B	Inference with LLMs
UniversalNER (Zhou et al., 2023b)	59.4	31.2	-	0	zero-shot	UniNER-7B	Inference with LLMs
GNER (Ding et al., 2024a)	68.6	47.5	-	0	zero-shot	GNER-LLAMA-7B	Inference with LLMs
YAYI-UIE (Xiao et al., 2023)	68.6	47.5	-	0	zero-shot	GNER-LLAMA-7B	Inference with LLMs
GOLLIE (Sainz et al., 2023)	68.4	52.7	-	0	zero-shot	Code-LLaMA-34B	Inference with LLMs
KnowCoder (Li et al., 2024b)	50.0	48.2	-	0	zero-shot	LLaMA2-base-7B	Inference with LLMs

Table 2: Performance of generative IE methods in NER task under various conditions.

Methods	Datasets			Training?	Experimental Setup	Model	Category
	ACE2005	Conl04	WebNLG				
Copymtl (Zeng et al., 2020)	-	-	60.5	1	full	-	-
Metaretriever (Yu et al., 2023)	64.37	73.66	-	1	full	T5	Prompt Learning
(Veyseh et al., 2023)	71.13	-	-	1	full		Data Manipulation
JoinER-BART (Chang et al., 2023)	-	-	91.37	1	full	BART	Decoding
REBEL (Cabot and Navigli, 2021)	-	75.4	-	1	full	BART	Training Task
REKnow (Zhang et al., 2022a)	68.3	-	87.0	1	full	T5	Prompt Learning
R-TES (Zhang et al., 2023c)	-	-	90.7	1	full	T5	-
RevisitingRE (Wadhwa et al., 2023)	-	80.76	-	1	full	T5	Prompt Learning
SetLearning-for-RE (Li et al., 2023c)	65.9	73.7	-	1	full	T5	Prompt Learning
LasUIE (Fei et al., 2022)	66.4	75.3	-	1	full	T5	Training Task/Model Architecture
IE-E2H (Gao et al., 2023)	66.40	75.31	-	1	full	T5	Training Task
ASP (Liu et al., 2022a)	72.7	76.3	-	1	full	T5	Decodin
Universal-IE (Lu et al., 2022b)	66.06	75.00	-	1	full	T5	Training Task
InstructUIE (Wang et al., 2023b)	82.31	78.48	-	1	full	FlanT5-11B	Training Task
YAYI-UIE (Xiao et al., 2023)	84.14	79.73	-	1	full	Baichuan2-13B	Training Task
GOLLIE (Sainz et al., 2023)	70.1	-	-	1	full	Code-LLaMA-34B	Training Task
KnowCoder (Li et al., 2024b)	64.5	73.3	-	1	full	LLaMA2-base-7B	Training Task
C-ICL (Mo et al., 2024)	22.31	56.93	-	0	zero-shot	CodeLlama-34B	Inference with LLMs
	FewRel	Wiki-ZSL					
RAG-ZSRTE (Zhang et al., 2023d)	27.07	32.75		1	low	BART	Data Manipulation
ZETT (Kim et al., 2022)	30.71	21.49		1	low	BART	Training Task
RelationPrompt (Chia et al., 2022)	30.01	22.34		1	low	BART	Data Manipulation
MetaLearning-for-ZSRT (Li and Qian, 2023)	39.15	36.56		1	low	T5	Training Task
InstructUIE (Wang et al., 2023b)	39.55	35.20		0	zero-shot	FlanT5 11B	Inference with LLMs
YAYI-UIE (Xiao et al., 2023)	36.09	41.07		0	zero-shot	GNER-LLAMA-7B	Inference with LLMs

Table 3: Performance of generative IE methods in RE task under various conditions.

Methods	Datasets			Training?	Experimental Setup	Model	Category
	ACE2005	SemEval2010	TACRED				
GREC (Ni et al., 2022)	70.2	89.9	80.6	1	full	BART/T5	Training Task
GenPT (Han et al., 2022)	-	-	75.3	1	full	BART/T5	Prompt Learning
RELA (Li et al., 2023a)	-	90.4	71.2	1	full	BART	Training Task
TANL (Paolini et al., 2021)	-	-	71.9	1	full	T5	Training Task
TAG (Triplets)	-	-	64.9	1	low	BART	-
SuRE (Lu et al., 2022a)	19.47(tacred)	39.27(semeval2010)	-	0	zero-shot	chatgpt	Training Task

Table 4: Performance of generative IE methods in RC task under various conditions.

Methods	Datasets			Training?	Experimental Setup	Model	Category
	RAMS	WIKIEVENTS	ACE05-E				
SPEAE (Nguyen et al., 2023)	58.0/53.3	71.9/66.1	-	1	full	Prompt Learning	
Universal-IE (Lu et al., 2022b)	-	-	54.79	1	full	T5	Training Task
IE-E2H (Gao et al., 2023)	-	-	53.85	1	full	T5	Training Task
LasUIE (Fei et al., 2022)	-	-	51.7	1	full	T5	Training Task/Model Architecture
TANL (Paolini et al., 2021)	-	-	48.5/48.5	1	full	T5	Training Task
Metaretriever (Yu et al., 2023)	-	-	52.62	1	full	T5	Training Task
DEGREE (Hsu et al., 2021)	-	-	73.3/55.8	1	full/low	BART	Prompt Learning
GEN-ARG (Li et al., 2021b)	-	72.29/65.11	-	1	full	BART/T5	-
Memory-Docie (Du et al., 2022)	-	64.31/58.78	-	1	full	BART	Prompt Learning
EEasQA (Lu et al., 2023a)	-	-	75.0/72.8	1	full	BART/T5	Training Task
Simple to Complex (Huang et al., 2023)	-	65.31/58.65	-	1	full	BART	Prompt Learning
KEPGEE (Song et al., 2023)	-	-	60.8/58.3	1	full	BART	Prompt Learning
PAIE (Ma et al., 2022)	56.8/52.2	70.5/65.3	75.7/72.7	1	full/low	BART	Prompt Learning
RGQA (Du and Ji, 2022)	-	-	75.51/72.75	1	full/low	BART	Training Task
Retrieve-and-Sample (Ren et al., 2023)	54.6/48.4	69.6/63.4	-	1	full	T5	-
Role Knowledge Prompting (Hu et al., 2023a)	55.1/50.3	69.1/63.8	-	1	full	BART	Prompt Learning
TEXT2EVENT (Lu et al., 2021)	-	-	-/53.8	1	full	T5	Training Task
Overlap (Zhang et al., 2023b)	58.1/54.3	73.7/68.7	78.2/75.4	1	full/low	BART	Prompt Learning
ChatGPT-IE (Han et al., 2023b)	25.09	-	-	0	zero-shot	chatgpt	Inference with LLMs
		ACE05-E					
DEGREE (Hsu et al., 2021)		63.3/57.3		1	low(10% training data)	BART	Prompt Learning
PAIE (Hsu et al., 2021)		55		1	low(10% training data)	BART	Prompt Learning
RGQA (Hsu et al., 2021)		52.55		1	low(9.5% training data)	BART	Prompt Learning
Overlap (Hsu et al., 2021)		59.3		1	low(200 training instances)	BART	Prompt Learning

Table 5: Performance of generative IE methods in EAE task under various conditions. If the value has two values, they represent Arg-I and Arg-C.

Methods	Datasets	Training?	Experimental Setup	Model	Category
	ACE05-E				
Universal-IE (Lu et al., 2022b)	73.36	1	full	T5	Training Task
IE-E2H (Gao et al., 2023)	72.19	1	full	T5	Training Task
ANL (Paolini et al., 2021)	71.8/68.5	1	full	T5	Training Task
Metaretriever (Yu et al., 2023)	72.38	1	full	T5	Training Task
KEPGEE (Song et al., 2023)	80.9/76.2	1	full	Bart	Prompt Learning
KiPT (Li et al., 2022a)	78.6/75.3	1	full	T5	Prompt Learning
KiPT (Li et al., 2022a)	63.6	1	low(64-shot)	T5	Prompt Learning
TEXT2EVENT (Lu et al., 2021)	-/71.9	1	full	T5	Training Task
ChatGPT-IE (Han et al., 2023b)	17.55	0	zero-shot	ChatGPT	Inference with LLMs
KnowCoder (Li et al., 2024b)	74.2	1	full	LLaMA2-base-7B	Training Task

Table 6: Performance of generative IE methods in ED task under various conditions. If the value has two values, they represent Trig-I and Trig-C.

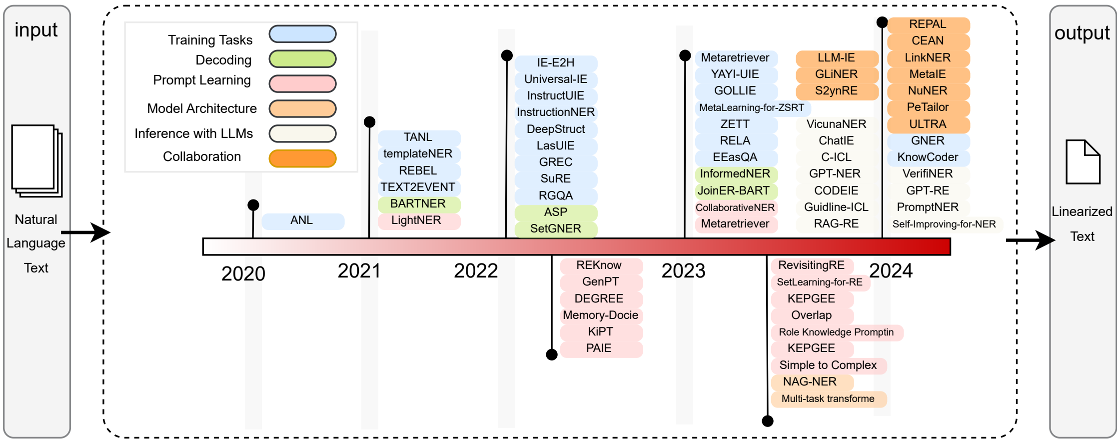


Figure 4: The development of generative IE methods. With the advent of large language models (LLMs), approaches based on LLM reasoning, as well as methods leveraging the collaboration between small language models (SLMs) and LLMs, have been steadily gaining traction. Meanwhile, methods relying on decoding, prompt learning, and model architecture have gradually declined in prominence. However, approaches centered on training tasks have consistently remained integral and uninterrupted throughout.

Figure 13, we have clarified the conversion process of output linearization.

G IE Examples

To gain an intuitive understanding of different IE tasks, we have provided an example for NER, RE, and EE in Figure 5. Figure 5a illustrates three common entity types in Named Entity Recognition (NER) tasks: flat entity types, nested entity types, and discontinuous entity types.

H The evolution of IE tasks

In this section, we will briefly review the evolution of NER, RE, and EE, respectively.

The classical methods of NER involve regarding it as sequence tagging task (Devlin et al., 2018). However, when it comes to complex NER (e.g., nested NER and discontinuous NER), elaborate tagging strategies are required (Ratinov and Roth, 2009; Wei et al., 2019; Alshammari and Alanazi, 2021; Beryozkin et al., 2019). Therefore, the span-based method is proposed (Li et al., 2021a; Su et al., 2022a). It attempts to assign an entity type to each subsequence which may be continuous or discontinuous. This approach naturally accommodates more complex NER tasks (e.g., nested NER and discontinuous NER). However, the span-based paradigm overlooks interactions between entities and leads to relatively high time complexity ($\mathcal{O}(N^2)$) (Shen et al., 2023). Furthermore, generative-based methods have emerged (as shown in Figure 6).

RE is more challenging than NER. Initially, researchers tend to first identify the entities and then determine the relationship between them (Chan and Roth, 2011; Vashishth et al., 2018; Miwa and Bansal, 2016). However, these methods lead to the accumulation of errors and ignore the interaction among relation triplets (Nayak and Ng, 2020; Giorgi et al., 2022; Kambar et al., 2022). To mitigate the above issues, researchers have proposed jointly extracting entities and relations, which employ parameter sharing to extract entities and predict their associated relation (Wei et al., 2019; Dixit and Al-Onaizan, 2019; Han et al., 2023a). However, these methods still resemble pipelines, with a separate decoding process, and may still lead to some degree of error propagation (Wei et al., 2019). In order to address the aforementioned problems, researchers have proposed a generative framework to complete RE tasks (Nayak and Ng, 2020) (as shown in Figure 7).

Compared to NER and RE, the target of EE is more complex. Similar to the evolution of RE, it initially identifies the event type and then recognizes the corresponding arguments (Chen et al., 2015; Subburathinam et al., 2019). Nevertheless, this paradigm may result in error accumulation (Li et al., 2022b), prompting researchers to suggest the joint-based paradigm. In the joint-based paradigm, first, the trigger words and arguments are recognized simultaneously. Then, assign the event type and argument roles to them (Nguyen et al., 2016; Li et al., 2013; Zhang et al., 2016; Lin et al., 2020). However, both pipeline-based and joint-based event extraction methods are inevitable due to the influence of errors in event-type prediction on argument extraction effectiveness (Li et al., 2022b). Therefore, generative-based methods have been proposed to alleviate the above problems (as shown in Figure 8).

I Overview of PLMs-based and LLMs-based methods of Generative IE

Figure 12 illustrates the differences between LLMs and PLMs. LLMs are a subset of PLMs distinguished by their superior language understanding and emergent capabilities (Zhao et al., 2023b). These emergent capabilities allow LLMs to perform IE tasks using prompts. In contrast, using PLMs for IE tasks typically requires pre-training or fine-tuning to achieve comparable performance. To have a clear overview of PLMs-based and LLMs-based methods of generative IE, we show the general framework in Figure 9, Figure 10.

J Decoding

A brief introduction to the three common decoding methods in generative IE is provided, as shown in Figure 11. Figure11a depicts the most common autoregressive decoding method. Figure11b illustrates the set decoding paradigm, where a query vector can generate a corresponding information extraction (IE) target. Figure 11c presents the constraint decoding paradigm, which modifies the probability distribution of the tokens predicted by the model.

K Model scaling in generative IE

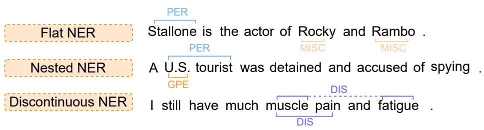
We examined whether the modeling scaling law (Kaplan et al., 2020) still exists in generative IE tasks. The experimental setup in Figure 2 follows (Han et al., 2023b). Specifically, we select the zero-shot prompt with the best performance in Han et al. (2023b) and add the randomly selected samples from the corresponding training set. The demonstration of ICL includes 5 examples. We run 5 experiments and then take the average value for reporting. The dataset used in the experiment comes from (Wang et al., 2023b), and the statistical results of the dataset are as shown in Table 8. The experimental models we have chosen are GPT2 base (124M), GPT-2 medium (335M), GPT-2 large (774M), GPT-2 xl (1.5B), and openllama-3B-v2 (3B), Llama-2-7B, Llama-2-13B, and Llama-2-70B.

Method	Abstract Instance	Specific Instance	Task
Natural Language Text Sequence			
(Cui et al., 2021)	<candidate_span> is a <entity_type> entity	**U.S. tourist** is a **person** entity.	NER
(Wang et al., 2022c)	<candidate_span> is a/an <entity_type>	**U.S. tourist** is a/an **person**.	NER
(Athiwaratkun et al., 2020)	-	A **[U.S. tourist | person]** was detained and accused of spying.	NER
(Ding et al., 2024a)	-	A (O) U.S. (B-PER) tourist (I-PER) was (O) detained (O) and (O) accused (O) of (O) spying (O) . (O)	NER
(Zhang et al., 2023c)	relation type of subject is object.	Nationality of Stephen Chow is China.	RE
(Song et al., 2023; Li et al., 2024a)	-	Event trigger is detonated. Palestinian attacked jeep and soldiers by bomb in Gaza Strip.	EE
Special Token Text Sequence			
(Iovine et al., 2022)	entity <sep> entity type <sep>	U.S. tourist <sep> person <sep>	NER
(He and Tang, 2022; Mo et al., 2023)	entity span <entity type> </s>	U.S. tourist <PER></s>	NER
(Chen et al., 2023b)	(entity type: entity)	(person: U.S. tourist)	NER
(Zhang et al., 2023e)	<s> <entity type> entity span </s>	<s> <PER> U.S. tourist </s>	NER
(Deußer et al., 2023)	entity type <TCS> entity span <ES>	person <TCS> U.S. tourist <ES>	NER
(Wang et al., 2023b)	(entity, entity type)	(U.S. tourist, person)	NER
(Xiao et al., 2023)	entity type: [entity]	person: [U.S. tourist]	NER
(Cao and Ananiadou, 2021; Nayak and Ng, 2020)	subject entity ; object entity ; relation type	Stephen Chow; China; Nationality	RE
(Huang et al., 2021)	<SOT><SOSN>subject entity type<EOSN><SOE>subject entity<EOE><SOSN>object entity type<EOSN><SOE>object entity<EOE><EOT>	<SOT><SOSN>person<EOSN><SOE>Stephen Chow<EOE> <SOSN>country<EOSN><SOE>China<EOE><EOT>	RE
(Cabot and Navigli, 2021)	<tirplet> subject entity <subj> object entity <obj> relation type	<tirplet> Stephen Chow <subj> China <obj> Nationality	RE
(Ni et al., 2022)	[subject entity | relation type | object entity]	[Stephen Chow | Nationality | China]	RE
(Giorgi et al., 2022)	subject entity @subject entity type@ object entity @object entity type@ @relation type@	Stephen Chow @person@ China @country@ @Nationality@	RE
(Chia et al., 2022; Tan et al., 2022)	Head Entity: subject entity, Tail Entity: object, Relation: relation type.	Head Entity: Stephen Chow, Tail Entity: China, Relation: Nationality.	RE
(Zhang et al., 2022a)	<subject entity, relation type, object entity>	<Stephen Chow, Nationality, China>	RE
(Zhang et al., 2023c)	object entity | subject entity | relation type	China | Stephen Chow | Nationality	RE
(Li et al., 2023c; Wang et al., 2023b)	(subject entity, relation type, object entity)	(Stephen Chow, Nationality, China)	RE
(Chen et al., 2024c)	_{subject entity} <rel> relation type </rel> <obj> object entity </obj>	_{Stephen Chow} <rel> Nationality </rel> <obj> China </obj>	RE
(Han et al., 2022)	[X] subject entity type [Y] tail entity type [Z] relation type [W]	[X] person [Y] country [Z] Nationality [W]	RC
(Kim et al., 2022)	[X] subject entity [Y] object entity [Z]	[X] Stephen Chow [Y] China [Z]	RE
(Li et al., 2023f)	[subject entity, relation type, object entity]	[Stephen Chow, Nationality, China]	RE
(Xiao et al., 2023)	[relation: relation type, head: subject entity, tail: object entity]	[relation: Nationality, head: Stephen Chow, tail: China]	RE
(Lu et al., 2021)	((type trigger words (role1 argument1) (role2 argument2)))	((Conflict:Attack detonated (Attacker Palestinian) (Target jeep) (Target soldiers) (Instrument bomb) (Place Gaza Strip)))	EE
(Li et al., 2022a)	trigger words <triggers> event type	detonated <triggers> Conflict:Attack	ED
(Wang et al., 2023b)	(type: event type, trigger: trigger words, argument role: argument)	(type: Conflict:Attack, trigger: detonated, Attacker: Palestinian, Target: jeep, Instrument: bomb, Place: Gaza Strip)	EE
(Xiao et al., 2023)	[trigger: trigger words, type: event type, arguments: role: name]	[trigger: detonated, type: Conflict:Attack, arguments: Attacker: Palestinian, Target: jeep, Instrument: bomb, Place: Gaza Strip]	EE
Code Text Sequence			
(Li et al., 2023e)	-	entity_list.append({"text": "U.S. tourist", "type": "person"}); entity_relation_list.append("rel_type": "Nationality", "ent1_type": "person", "ent1_text": "Stephen Chow", "ent2_type": "country", "ent2_text": "China")	NER; RE
(Sainz et al., 2023)	-	result=[Person(span="U.S. tourist")]	NER

Table 7: The examples of the linearized text. **For NER**, the entity type is *PER*, and the entity is *U.S. tourist*. **For RE**, the subject entity is *Stephen Chow*, the object entity is *China*, and the relation type is *Nationality*. **For EE**, the label is: event type: *Conflict:Attack*; trigger words: *detonated*; Attacker: *Palestinian*; Target: *jeep, soldiers*; Instrument: *bomb*; Place: *Gaza Strip*.

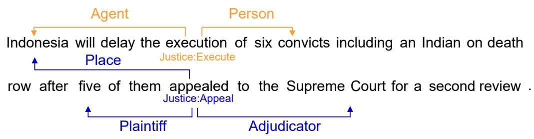
Task	Dataset	#Train	#Val	#Test
NER	CoNLL2003	14041	3250	3453
NER	Ace2005	7299	971	1060
RE	CoNLL2004	922	231	288
ED	Ace2005	3342	327	293
EAE	Ace2005	3342	327	293

Table 8: Dataset statistics.



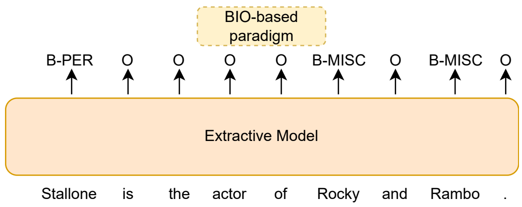
(a) Examples of NER include flat NER, nested NER, and discontinuous NER.

(b) The example of RE. The arrow indicates the direction from the subject entity to the object entity, with the corresponding relation type annotated on the arrow.

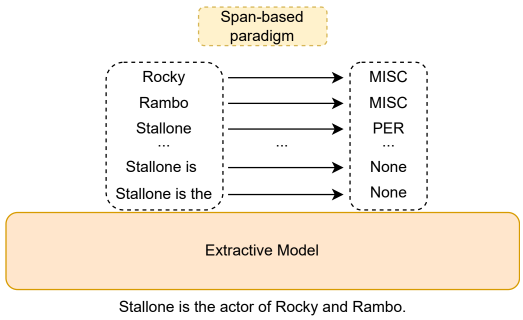


(c) The example of EE. The arrow indicates the direction from the trigger word to the argument, with the corresponding argument role annotated on the arrow.

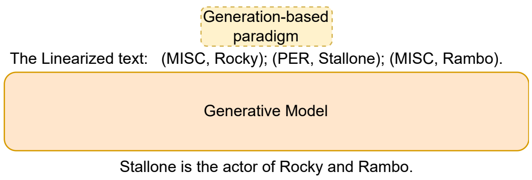
Figure 5: IE examples.



(a) The BIO-based paradigm assigns a BIO tag to each word in the source sentence. The paradigm has difficulty handling complex NER tasks (e.g., nested NER, discontinuous NER).

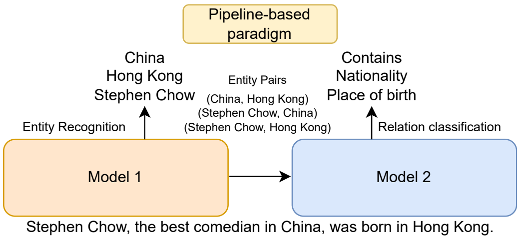


(b) The span-based paradigm attempts to enumerate all available sub-sequences within the source sentence and assigns an entity type to each sub-sequence. The sub-sequences might be continuous or discontinuous. The paradigm naturally accommodates more complex NER tasks such as nested NER and discontinuous NER.

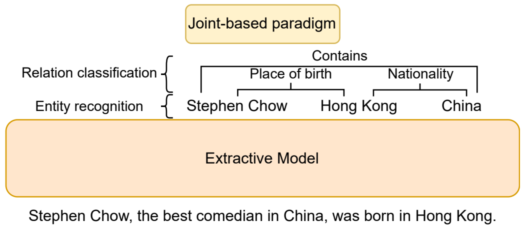


(c) The generation-based paradigm reformulates NER as the generation task and generates the linearized text.

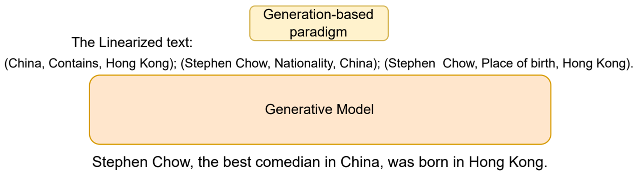
Figure 6: The evolution of NER



(a) The pipeline-based paradigm first identifies entities using the NER system, followed by employing a classifier to determine the relation among them. It will lead to the accumulation of errors and ignore the interaction among relation triplets within the source sentence.

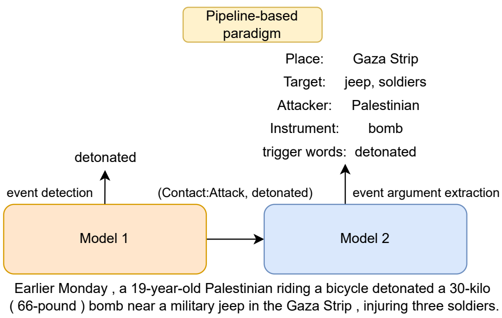


(b) The joint-based paradigm uses a model to first identify entities and then perform relation classification. It still resembles pipelines, with a separate decoding process, and may still lead to some degree of error propagation.

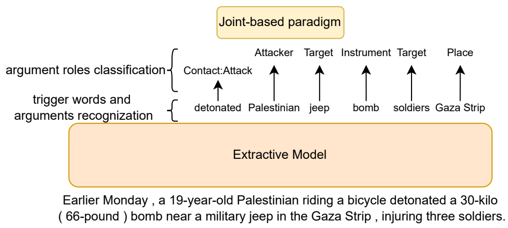


(c) The generation-based paradigm reformulates RE as the generation task and generates the linearized text.

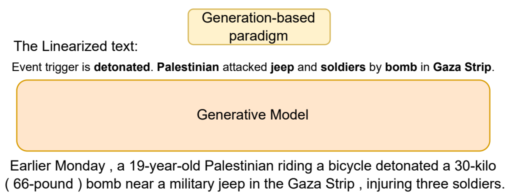
Figure 7: The evolution of RE



(a) The pipeline-based paradigm is necessary to identify the event type and the trigger words in the source sentence, and then, given the event type, the corresponding event arguments are extracted



(b) The joint-based paradigm recognizes the triggers and arguments simultaneously in the first stage. In the second stage, to avoid the error information propagation from event type, trigger classification, and argument role classification are realized simultaneously



(c) The generation-based paradigm reformulates EE as the generation task and generates the linearized text.

Figure 8: The evolution of EE

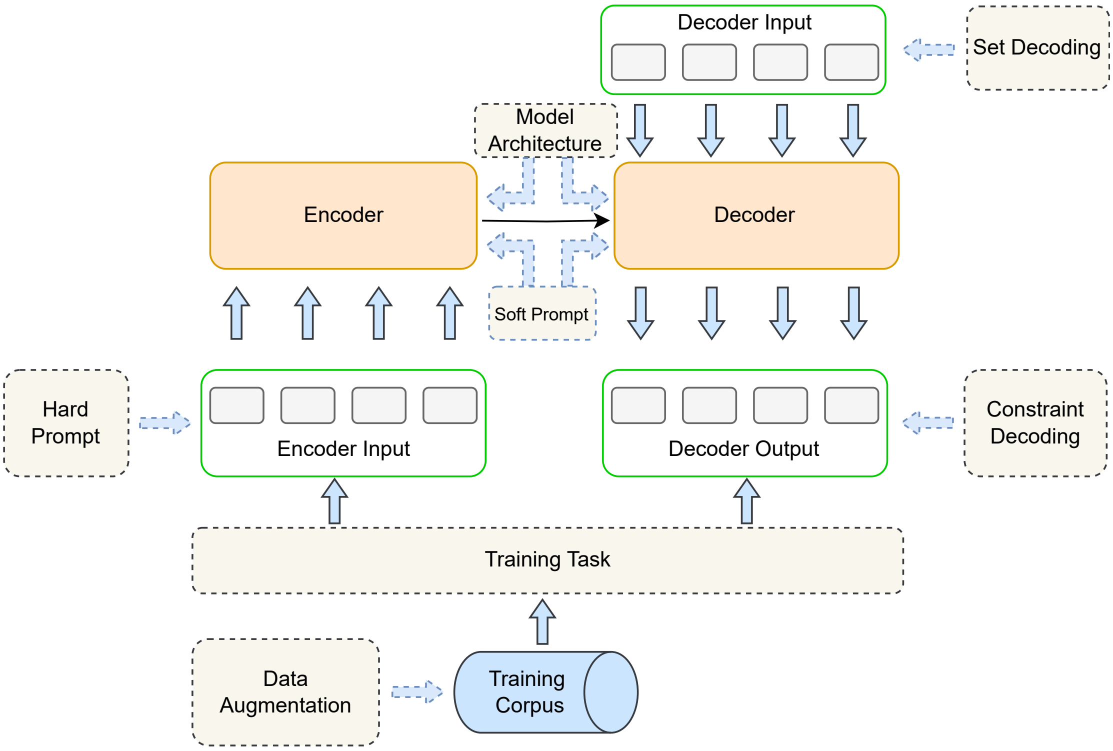


Figure 9: The overall framework of PLMs for generative IE. The dashed box indicates the technologies that can be applied to the PLMs. And The dashed arrow indicates that the techniques in this part can be selectively applied to PLMs.

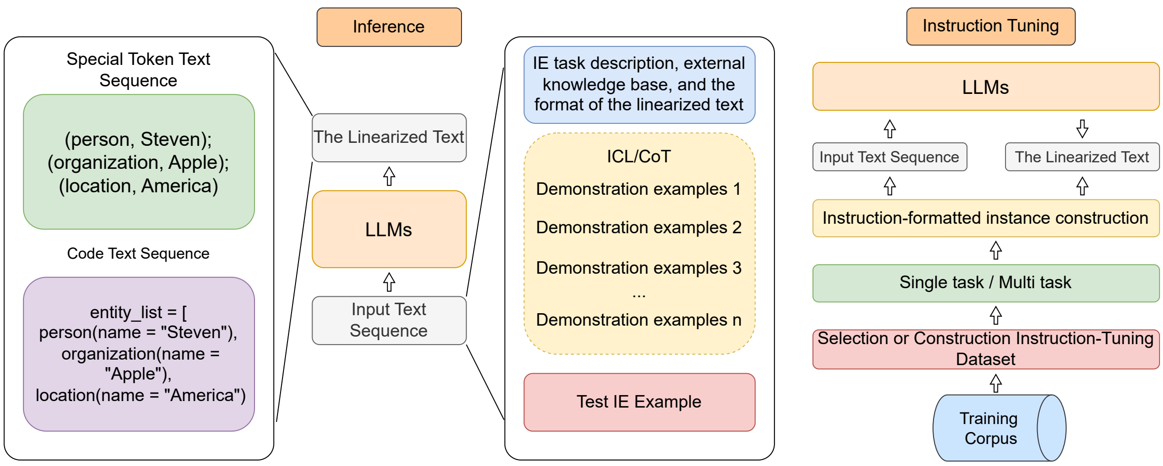
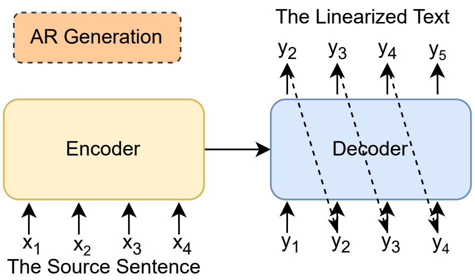
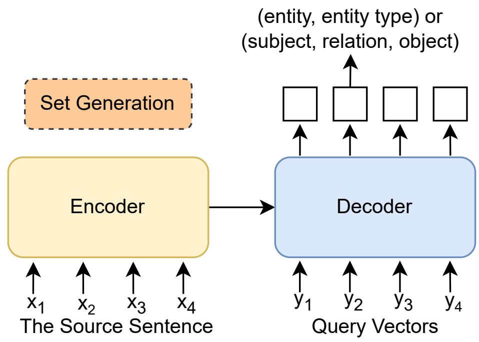


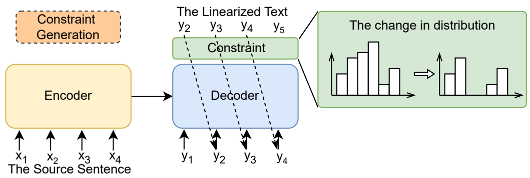
Figure 10: The overall framework of LLMs for generative IE.



(a) In the generative process of the autoregressive paradigm, the linearized text tokens are sequentially generated one by one.



(b) In the set generation, the decoder receives query vectors as input, with each query vector being decoded into an element of the IE target set.



(c) In the constraint generation, the vocabulary distribution of each token needs to be constrained before it is generated. After the constraints, the distribution of token has changed.

Figure 11: Decoding

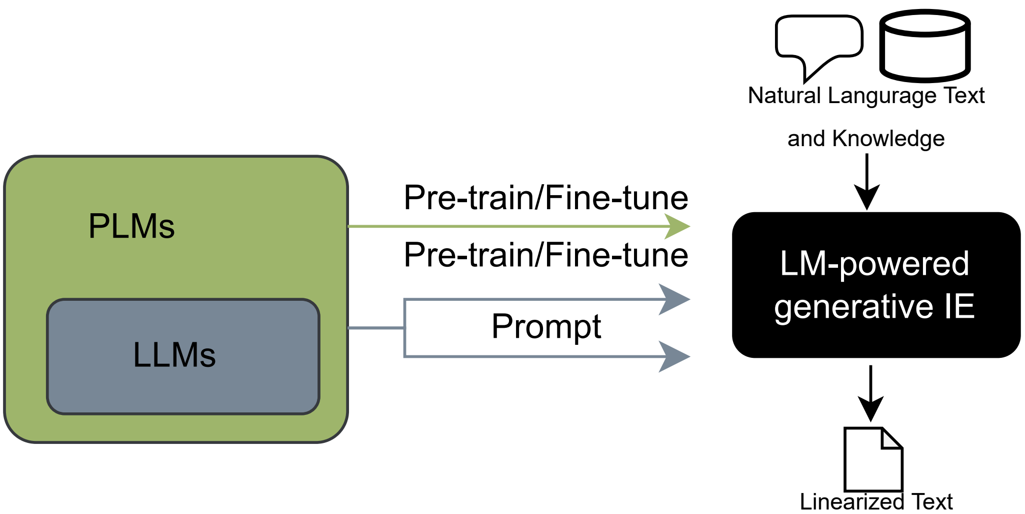


Figure 12: The overview of PLMs-based and LLMs-based methods of Generative IE.

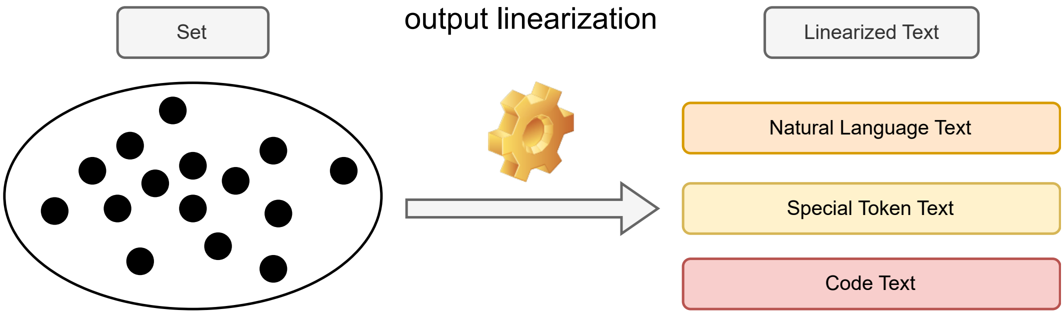


Figure 13: A clear presentation of the output linearization process.