AgentRE: An Agent-Based Framework for Navigating Complex Information Landscapes in Relation Extraction

Yuchen Shi
ycshi21@m.fudan.edu.cn
School of Data Science, Fudan University
Shanghai, China

Guochao Jiang
gcjiang22@m.fudan.edu.cn
School of Data Science, Fudan University
Shanghai, China

Tian Qiu
tqiu22@m.fudan.edu.cn
School of Data Science, Fudan University
Shanghai, China

Deqing Yang✉
yangdeqing@fudan.edu.cn
School of Data Science, Fudan University
Shanghai, China

Abstract

The relation extraction (RE) in complex scenarios faces some challenges such as diverse relation types and ambiguous relations between entities within a single sentence, leading to the poor performance of pure "text-in, text-out" language models (LMs). To address these challenges, in this paper we propose an agent-based RE framework, namely *AgentRE*, which employs a large language model (LLM) as the agent interacting with some modules to achieve complex RE tasks. Specifically, three major modules are built in AgentRE serving as the tools to help the agent acquire and process various useful information, thereby obtaining improved RE performance. Our extensive experimental results upon two datasets in English and Chinese, respectively, demonstrate our AgentRE's superior performance, especially in low-resource scenarios. Additionally, the trajectories generated by AgentRE can be refined to construct a high-quality training dataset incorporating different reasoning methods, which can be used to fine-tune smaller models.[1]

CCS Concepts

• **Computing methodologies** → **Information extraction**.

Keywords

relation extraction, agent, large language model, retrieval, memory

---

[1]Code is available at https://github.com/Lightblues/AgentRE.

---

1 Introduction

Relation extraction (RE) aims to transform unstructured text into structured information (relational triple), and plays a pivotal role in many downstream tasks, including semantic understanding and knowledge graph (KG) construction [1, 17]. However, some challenges such as the diversity of relation types and the ambiguity of relations between entities in a sentence [28, 35], often hinder the models of "**text-in, text-out**" scheme[2, 34] from achieving effective RE.

In recent years, large language models (LLMs) have demonstrated powerful capabilities including natural language understanding and generation, and thus been widely employed in many tasks [30, 37, 38]. There have been some efforts employing LLMs to achieve information extraction tasks, through converting structured extraction tasks into sequence-to-sequence tasks of natural language generation. These approaches usually adopt natural language or code to describe relation schemata [8, 27]. Despite of their advancements, these approaches are often restricted to supervised fine-tuning [16, 27] or few-shot QA-based extraction [12, 29, 36], less exploring LLMs' potential in complex RE scenarios.

It is worth noting that, employing LLMs to achieve the RE tasks in complex scenarios has to face several challenges as follows:

1. *How to utilize LLMs's capabilities to better leverage various significant information related to RE?* There exists various information, such as labelled samples, the articles and the knowledge from KGs related to the objective relations, that can be leveraged by RE models to improve RE performance. However, the limited context window of LLMs hinders the full utilization of comprehensive significant information.

2. *How to leverage LLMs to achieve RE effectively in specific or low-resource domains?* Many specific domains only have sparse data, making traditional supervised models difficult to obtain satisfactory performance.

3. *How to achieve effective RE with affordable costs?* Although LLMs have better performance, relatively smaller models are still considerable in practise for their affordable computational resource consumption. Thus, using the knowledge distilled from larger models to fine-tune smaller models is a reasonable way.

Previous works [22, 26] have demonstrated that, the agent-based framework can endow LLMs with more capabilities such as memory, reflection and interaction with outside environment, thereby

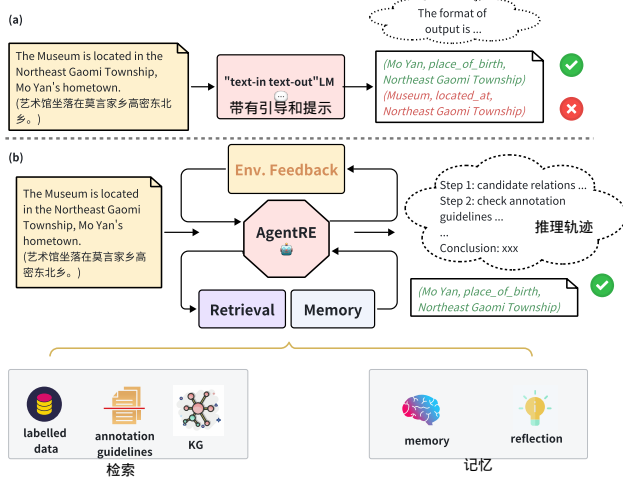Yuchen Shi, Guochao Jiang, Tian Qiu, and Deqing Yang ✉

Figure 1: Subfigure (a) illustrates the RE process of a language model of "text-in, text-out" scheme, which generates the results with errors directly from the input text or through simple prompting methods . Subfigure (b) illustrates the RE process of our proposed AgentRE, which is an agent-based framework including the retrieval and memory modules, and utilizes various information during multiple reasoning rounds to achieve more accurate RE.

facilitating the achievement of complex RE. Inspired by them, in this paper we propose a novel agent-based framework for RE, namely **AgentRE**, which addresses the aforementioned challenges as follows.

Firstly, to better leverage various significant information in complex contexts, AgentRE employs the LLM as an agent and processes the data from various sources. It utilizes some tools such as retrieval and memory module to aid the agent's reasoning process. For instance, as illustrated in Figure 1, unlike conventional "text-in, text-out" LMs relying on single-round input-output to achieve RE, AgentRE engages in multiple rounds of interaction and reasoning. This approach enables the utilization of a broader spectrum of information sources for extraction tasks, avoiding the limitations in single-round extraction.

Secondly, facing the situations of low-resource, AgentRE can make dynamic summarizations and reflections throughout the extraction process with the help of the LLM's reasoning and memory capability. As a result, AgentRE is adept at continual learning, improving its extraction capability through an ongoing process of summarizing experiences and accumulating knowledge.

Finally, we introduce a method for converting the reasoning trajectories of AgentRE into high-quality data, which encompass various reasoning strategies such as direct generation, step-by-step extraction, and CoT (Chain-of-Thought) based extraction. The enriched data can be utilized to fine-tune relatively small models, guiding them to dynamically select different extraction methods (as discussed in [4]), thereby enhancing the small models' extraction performance.

In summary, the main contributions of this paper include:

1. We propose an agent-based RE framework *AgentRE*, in which the agent can explore and collect more significant information to improve RE, with the retrieval, memory and extraction modules.

2. Our extensive experiments on two datasets in English and Chinese not only validate AgentRE's state-of-the-art (SOTA) performance in low-resource RE tasks, but also verify the effectiveness of each module built in AgentRE.

3. By utilizing the reasoning trajectories of the agent in AgentRE, the refined records can be utilized to construct a dataset incorporating diverse reasoning methods. Through distillation learning, the reasoning-based extraction capabilities can be transferred from large models to relatively small models, to achieve satisfactory RE with affordable costs.

2 Related Work

2.1 LLM-based Information Extraction

Recent studies [2, 8, 27, 29] have explored using LLMs for information extraction (IE). The research can be categorized into two groups. The first group focuses on LLMs designed for specific IE tasks, such as named entity recognition (NER) [39], relation extraction (RE) [36], and event extraction (EE) [40]. These models often perform better but require separate fine-tuning for each task. The second group aims to handle multiple IE tasks with a single model, creating a universal extraction model [8, 16, 27]. This approach uses a unified method with designed prompts to address various tasks, enhancing generalization but sometimes underperforming on specific tasks [32].

Furthermore, CooperKGC [34] has tried to utilize agents to tackle diverse IE subtasks. It emphasizes information interaction among multiple agents, using individual agents for different subtask. In contrast, our paper explores various types of information sources that could be utilized in IE tasks, with a stronger focus on leveraging agent memory and reasoning to accomplish extraction in complex scenarios.

2.2 LLM-based Agent

In recent years, LLM-based agents have gained significant attention. LLMs demonstrate strong task-solving and reasoning capabilities for both real and virtual environments. These abilities resemble human cognitive functions, enabling these agents to perform complex tasks and interact effectively in dynamic settings.

Plannning: It involves the ability to strategize and prepare for future actions or goals. AUTOACT [18] introduces an automatic agent learning framework for planning that does not rely on large-scale annotated data and synthetic trajectories from closed-source models (e.g., GPT-4).

Tool Use: This is the capacity to employ objects or instruments in the environment to perform tasks, manipulate surroundings, or solve problems. KnowAgent [41] introduces a novel approach designed to enhance the planning capabilities of LLMs by incorporating explicit action knowledge.

Embodied Control: It refers to an agent's ability to manage and coordinate its physical form within an environment. This encompasses locomotion, dexterity, and the manipulation of objects. RoboCat [3] introduces a visual goal-conditioned decision transformer capable of consuming action-labeled visual experience.

Communication: It is the skill to convey information and understand messages from other agents or humans. Agents with advanced communication abilities can participate in dialogue, collaborate with others, and adjust their behaviour based on the communication received. Ulmer et al. [24] introduce an automated way to measure the partial success of a dialogue, which collects data through LLMs engaging in a conversation in various roles.

In this paper, our proposed AgentRE is built based an agent interacting with the environment, which primarily utilizes the capabilities of LLMs to achieve the RE in complex scenarios.

3 Proposed Method

3.1 Overview

The overview of our proposed framework is illustrated in Figure 2(a), where the core LLM-based agent plays the important role of reasoning and decision-making. The three modules around the agent, i.e., the *retrieval* module, *memory* module, and *extraction* module, serve as the tools to aid the agent on acquiring and processing information. We briefly introduce the functions of the three modules as follow.

Retrieval Module. It maintains relatively static knowledge to facilitate storing and retrieving information, including the annotated samples from the training dataset and the related information such as annotation guidelines.

Memory Module. It maintains relatively dynamic knowledge, including *shallow memory* for recording extraction results and *deep memory* for summarizing and reflecting on historical actions. Our framework records and utilizes the extraction experiences by reading from and writing to the memory module.

Extraction Module. It extracts structured information (triples) from the input text with various reasoning methods, based on the information provided by the retrieval and memory module.

Next, we introduce the design details of all modules in AgentRE.

3.2 Retrieval Module

The retrieval module in our framework serves as a critical component to source relevant samples from existing datasets and supplementary knowledge from various resources, and thus helps the extraction module achieve RE task. The retrievable data may be extensive and diverse, which, for the purpose of clarity, is categorized into two main types in this paper.

1. Labelled data with a clear input-output relationship $x \rightarrow y$, which can be organized into the context of the LLM as the few-shot examples, helping the model quickly understand the input-output relationship of the current task.

2. Other relevant information, such as relation descriptions, annotation guidelines, and even external knowledge in encyclopedia. By injecting them as aside information into the context of the LLM, they can assist the model on understanding the extraction task.[2]

---

[2]For a fair comparison with existing models, in our experiments our AgentRE does not leverage external web knowledge such as encyclopedia sites. However, existing work [42] has conducted the experiments in such a setting.

To effectively manage and utilize these two types of data, we introduce two specific retrieval modules: the *sample retrieval module* and the *relevant information retrieval module*. Once informative labelled data and other pertinent information are acquired, the retrieval module can leverage these insights. A straightforward approach is to concatenate them into prompts, thereby assimilating this beneficial information. The template of these prompts is depicted in Figure 3. It is worth mentioning that the extraction module may adopt various reasoning methods other then straightforward prompting, as detailed in Section 3.4.

3.2.1 Sample Retrieval. The sample retrieval module, as shown in the lower part of Figure 2(b), encodes the current text into an embedding with an encoder. It then calculates the similarities between the samples in the training dataset to retrieve the samples similar to the current text. For instance, for the sentence *"On May 9th, Nobel laureate and writer Mo Yan delivered a speech in Beijing."*, the sample retrieval module can retrieve relevant samples from the training dataset through embedding matching, such as the text *"When the newly minted Nobel Prize in Literature, British novelist Kazuo Ishiguro, found himself..."* with its corresponding label (relational triple) as *(Kazuo Ishiguro, award, Nobel Prize in Literature)*.

Specifically, the sample retrieval module includes a pretrained text encoder for converting input text into embedding, and an embedding retriever for retrieving the samples similar to the input text from the training dataset. Given the current input text x, it is encoded into an embedding $\mathbf{e}_x$, just like all samples $\{t_1, t_2, ..., t_N\}$ in the training dataset, as follows:

$$\mathbf{e}_x = \text{Encoder}(x), \tag{1}$$

$$\mathbf{e}_{t_i} = \text{Encoder}(t_i), \quad i = 1, 2, ..., N. \tag{2}$$

For all sample embedding set $\mathbf{E} = \{\mathbf{e}_{t_1}, \mathbf{e}_{t_2}, ..., \mathbf{e}_{t_N}\}$ constructed from the training data, the similarity between the input text embedding $\mathbf{e}_x$ and each sample embedding $\mathbf{e}_{t_i}$ can be calculated as cosine similarity, thus to obtain a similarity vector $\mathbf{s} = \{s_1, s_2, ..., s_N\}$ where $s_i = \text{cosine}(\mathbf{e}_x, \mathbf{e}_{t_i})$. Based on these similarity scores, the k most similar samples to the input text are retrieved. In fact, such an embedding retrieval process can be implemented through a standard retriever as

$$\{t_{i_1}, t_{i_2}, ..., t_{i_k}\} = \text{EmbeddingRetriever}(\mathbf{e}_x, \mathbf{E}, k), \tag{3}$$

where $\mathbf{E}$ is the embedding set and $\{i_1, i_2, ..., i_k\}$ represents the retrieved samples' positions in training dataset.

Additionally, when facing a large number of relation types, the extraction process might be decomposed into two distinct phases: identifying potential relation types presenting in the sentence, and then conducting the extraction based on these identified candidate relation types. The process of retrieving candidate relation types is represented by the dashed arrow in Figure 2 (b). A feasible approach for this retrieval is to develop a classifier trained on the dataset to predict the relations most likely to be found in the given text. Furthermore, the task of retrieving relation types can also be achieved using the inferential capabilities of LLMs, as discussed in Section 3.4.

3.2.2 Relevant Information Retrieval. The relevant information retrieval module, as shown in the upper part of Figure 2(b), is used to retrieve knowledge related to the given sentence. Compared to the

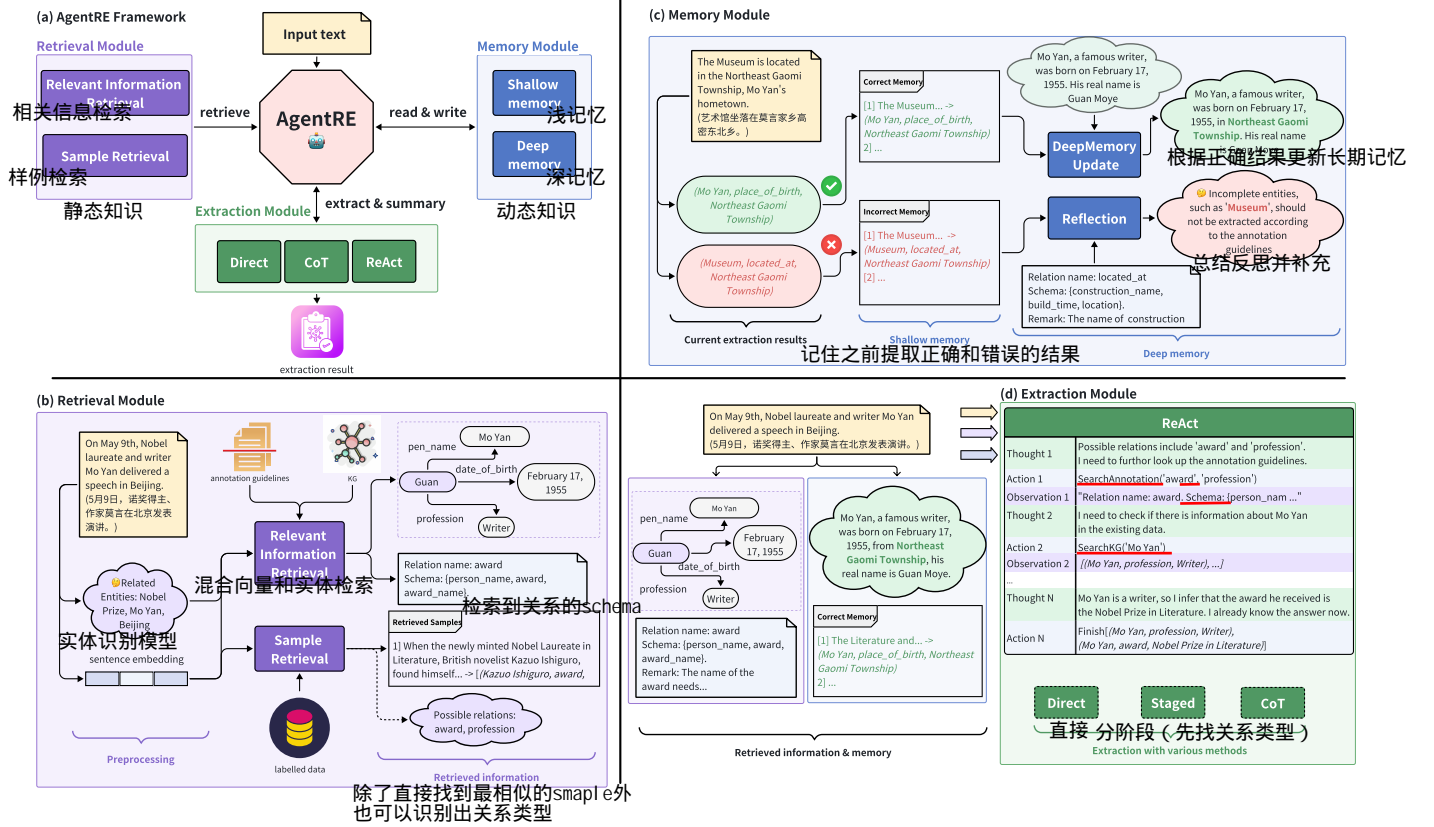Yuchen Shi, Guochao Jiang, Tian Qiu, and Deqing Yang ✉

Figure 2: The overview of our proposed framework AgentRE. Subfigure (a) depicts the overall structure of AgentRE, where the LLM acts as an agent to extract relation triples from the input text through the collaboration with the retrieval, memory, and extraction module. Subfigures (b)~(d) illustrate the design of the retrieval, memory, and extraction Module, respectively.

Task: Extract relation triples from the input text.
Input:
{input sentence x}
Some relevant examples:
{k retrieved examples}
Possible relation types:
{l retrieved relation}
Relevant information:
{retrieved relevant information}
Output:

Figure 3: The prompt template for the retrieval module.

embedding retrieval method used in Sample Retrieval, this module employs a variety of retrieval methods mixing vectors and entities to combine precise matching and fuzzy semantic matching.

For example, for the same sentence *"On May 9th, Nobel laureate and writer Mo Yan delivered a speech in Beijing."*, besides leveraging the sentence's representation, this module also identifies potential entities in the sentence, such as *Mo Yan*, *Nobel Prize* and *Beijing*,

and retrieves related knowledge using these entities. Additionally, based on the entity *Nobel Prize*, explanatory information about the candidate relation type *award*, including the definition of the head and tail entities of this relation type and detailed explanations, can be retrieved together from the annotation guidelines.

Formally, the relevant information retrieval module includes the preprocessing part of extracting key information or constructing embeddings, and several retrievers for retrieving information related to the input text. In the preprocessing part, besides the text encoder, there is also an Entity Recognizer for identifying all potential entities in the input text as

$$\{c_1, ..., c_{C_x}\} = \text{EntityRecognizer}(x), \tag{4}$$

where C_x is the number of entities identified in the input text x. In the retriever part, various methods can be used to retrieve related knowledge from different data sources, such as retrieving the attributes and relations of the entities from knowledge graph, retrieving explanatory information about the relations from annotation guidelines, or even retrieving related knowledge from external encyclopedias.

Besides the embedding-based retriever introduced above, here we introduce an entity-based retriever for retrieving knowledge related to the input text from existing KG. It mainly includes Entity

Linking and Entity Property Retrieval parts. Given a candidate entity mention c_i, we have

$$e_i = \text{EntityLinking}(c_i), \tag{5}$$

$$\{t_i^1, t_i^2, ..., t_i^{T_i}\} = \text{EntityPropertyRetrieval}(e_i), \tag{6}$$

where e_i is the entity linked by the entity linker from mention c_i, and $\{t_i^1, t_i^2, ..., t_i^{T_i}\}$ represents the triples related to entity e_i in the KG.

3.3 Memory Module

The roles of the memory module in AgentRE include dynamically utilizing existing knowledge during the extraction process, reflection and summarization, which helps AgentRE better achieve subsequent extraction tasks. Mimicking the human brain, the model's memory can be divided into *shallow memory* and *deep memory*.

3.3.1 Shallow Memory. Shallow memory refers to the preliminary records of extraction experiences. For example, as illustrated in Figure 2(c), for the sentence *"The Museum is located in Northeast Gaomi Township, Mo Yan's hometown."*, the model's extraction results are *(Mo Yan, place_of_birth, Northeast Gaomi Township)* and *(Museum, located_at, Northeast Gaomi Township)*. The first triple is correct but the second triple is marked as incorrect, due to the unclear referent of the mention *Museum*. In shallow memory, by recording the correct and incorrect results, the model can use them as the references in subsequent extractions. This process can be understood as the lessons learned from previous experiences. Specifically, the model adds a new record in **correct memory** and **incorrect memory**, respectively.

Formally, for an input sentence x, the extraction module generates M triples, denoted as $\hat{Y} = \{y_1, y_2, ..., y_M\} = \text{TripleExtractor}(x)$, where $y_i = (h_i, r_i, t_i)$ represents the i-th triple. After verifying each triple denoted as $\text{verify}(y_i)$, the correct triple set $Y_{\text{correct}} = \{y_i | y_i \in \hat{Y}, \text{verify}(y_i) = \text{True}\}$ and the incorrect triple set $Y_{\text{wrong}} = \{y_i | y_i \in \hat{Y}, \text{verify}(y_i) = \text{False}\}$ are obtained. Then, they are added into the memory component $\mathcal{M}_{Correct}$ or $\mathcal{M}_{Wrong}$ as

$$\mathcal{M}_{Correct} = \mathcal{M}_{Correct} \cup Y_{\text{correct}}, \tag{7}$$

$$\mathcal{M}_{Wrong} = \mathcal{M}_{Wrong} \cup Y_{\text{wrong}}. \tag{8}$$

3.3.2 Deep Memory. Deep memory includes the reflections and updates to historical memories, as shown in Figure 2(c). In deep memory, AgentRE can *update* long-term memories based on correct results and *reflect* on incorrect ones. Taking the example shown in Figure 2(c), given current correct extraction result, AgentRE updates its memory on entity *Mo Yan* from *"Mo Yan, a famous writer, was born on February 17, 1955. His real name is Guan Moye"* to a new one *"Mo Yan, a famous writer, was born on February 17, 1955, in Northeast Gaomi Township. His real name is Guan Moye."*. Moreover, for incorrect results, AgentRE performs reflection. For example, given an incorrect extraction result and relevant annotation guidelines, it generates the reflection text *"Incomplete entities, such as* Museum, *should not be extracted according to the annotation guidelines"*. Thus, if the next input text is *"The Museum, named after the most influential contemporary writer and scholar Mr. Wang Meng..."*, AgentRE can avoid similar errors by referring to previous reflections.

Formally, given an input sentence x and its correct extraction results Y_{correct}, AgentRE leverages each record (triple) $y_i \in Y_{\text{correct}}$ to update the deep memory $\mathcal{M}_{Deep}$ as

$$\mathcal{M}_{Deep} = \text{UpdateDeepMemory}(\mathcal{M}_{Deep}, y_i). \tag{9}$$

The update operation $\text{UpdateDeepMemory}(,)$ includes the following three steps:

$$m_i = \text{MemoryRetrieval}(\mathcal{M}_{Deep}, y_i), \tag{10}$$

$$m_i' = \text{MemoryUpdate}(m_i, y_i), \tag{11}$$

$$\mathcal{M}_{Deep} = \mathcal{M}_{Deep} \setminus \{m_i\} \cup \{m_i'\}. \tag{12}$$

Here, m_i and m_i' respectively represent the retrieved original memory and the updated memory. It should be noted that when the retrieved memory is empty, i.e., no related description is found, the model directly summarizes and adds the correct result into the deep memory.

For incorrect extraction results Y_{wrong}, the model reflects on each record $y_j \in Y_{\text{wrong}}$ and records the reflection outcome in the reflection memory as below,

$$r_j = \text{Reflection}(y_j), \tag{13}$$

$$\mathcal{M}_{Ref} = \text{UpdateRefMemory}(\mathcal{M}_{Ref}, r_j), \tag{14}$$

where r_j is the reflection result for the incorrect record y_j, and $\mathcal{M}_{Ref}$ denotes the reflection memory. Operation $\text{UpdateRefMemory}(,)$ includes recalling and updating related reflection memories, similar to the update operations for deep memory in Equation 9.

3.4 Extraction Module

We now present the overall extraction pipeline of extraction module in AgentRE. It adopts an interactive process similar to ReAct [33], engaging in multiple rounds of *Thought, Action, Observation*, as illustrated in Figure 2(d).

In this context, the retrieval and memory module are uniformly considered as the external tools used by the agent. As a series of APIs, the agent is provided with the tool name, input parameters when using these tools, and then receives the results. It allows the agent to dynamically decide *whether to call tools, which tools to call, and how to call them*.

For instance, still consider the sentence in Figure 2(d) *"On May 9th, Nobel laureate and writer Mo Yan delivered a speech in Beijing."*. In the first round, the agent identifies the potential relation types and then chooses to call the *SearchAnnotation* API to obtain relevant information. In the second round, the agent uses the *SearchKG* API to retrieve existing knowledge about *Mo Yan*. Finally, after gathering sufficient information, the agent executes the *Finish* action to return the extraction results.

It is important to note that, as shown in Figure 2(d), during extraction process, AgentRE may not always follow a complete multi-round ReAct interactions. Instead, it dynamically selects the appropriate extraction method based on the complexity of the input text. For example, it may use *Direct* extraction where the predicted relational triples are output directly from the input text, or *Staged* extraction where the relation types are first filtered, followed by the extraction of triples, or *Chain-of-Thought* (CoT) extraction where the final extraction results are generated step-by-step.

3.5 Distillation for Smaller Models

In the real-world applications, employing LLMs with robust reasoning capabilities as agents to achieve extraction tasks, has to face the problem of high costs. On the other hand, (relatively) smaller large language models (SLLMs) often exhibit comparatively weaker reasoning abilities. To bridge this gap, we introduce a distillation learning approach that leverages the historical reasoning trajectories of larger models to guide the learning of smaller models.

Prior research [4] has shown that applying diverse reasoning strategies to different types of problems can significantly improve a model's problem-solving versatility. For instance, in the context of RE tasks, straightforward relations that are explicitly mentioned in the text can be directly inferred to produce structured outputs. For the sentences encapsulating more complex relations, employing a CoT-based reasoning approach can guide the model through a step-by-step process towards the final result, thereby minimizing errors. Our AgentRE's reasoning framework, as described above, effectively employs tailored reasoning methodologies for varied scenarios through the agent. To endow SLLMs with similar capabilities while simplifying the reasoning process, we propose to distill more simplified rationales from AgentRE's historical reasoning trajectories, which are utilized to direct the learning of smaller models.

Formally, the sequence of thought, action and observation generated by AgentRE can be encapsulated into the following reasoning trajectory as

$$P = \left\{ p_j = (t_j, a_j, o_j) \right\}_{j=1}^{|P|}, \tag{15}$$

where t_j is the thought in the j-th iteration, a_j denotes the action taken, and o_j represents the observation, with the sequence extending over $|P|$ iterations. Integrating the reasoning trajectory with the input text and the accurate extraction results, allows the LLM to summary a more succinct rationale as

$$\{r_i, y_i\} = \text{Summarize}(P, x_i, y_i), \tag{16}$$

where r_i represents the summarized rationale, and y_i represents the correct extraction result. Such rationales can serve as the learning objectives for SLLMs, guiding their learning through supervised learning.

The accumulated extraction results with the rationales can be used to generate a novel training dataset $D' = \{(x_i, r_i, y_i)\}_{i=1}^{N}$, where N is the total sample number. This dataset enriches the original training dataset $D = \{(x_i, y_i)\}_{i=1}^{N}$ with the agent's distilled reasoning experiences, incorporating adaptive reasoning strategies. The objective of distillation learning with this enriched dataset is to empower SLLMs to select the most fitting reasoning approach based on the nuances of the input sentence. This learning process (supervised fine-tuning) can be formalized as

$$\theta'_{SLLM} = \text{SFT}(\theta_{SLLM}, D'), \tag{17}$$

where θ_{SLLM} and θ'_{SLLM} denote the initial and fine-tuned parameter set of the SLLM, respectively.

4 Experiments

4.1 Dataset Description

We have conducted extensive experiments to validate the effectiveness of AgentRE on the following two datasets.

DuIE [11][3] is the largest Chinese RE dataset, comprising 48 predefined relation types. Besides traditional simple relation types, it also includes complex relation types involving multiple entities. The annotated corpus was sourced from Baidu Baike, Baidu Information Stream, and Baidu Tieba texts, encompassing 210,000 sentences and 450,000 relations.

SciERC [17][4] is an English dataset for NER and RE in the scientific domain. The annotated data were derived from the *Semantic Scholar Corpus*, covering abstracts of 500 articles. The SciERC dataset includes 8,089 entities and 4,716 relation records in total, with an average of 9.4 relations per document.

4.2 Comparison Models

We compared our AgentRE with several LLM-based IE models/frameworks in our experiments as follows.

1) **ChatIE** [29] introduces a zero-shot IE approach through the dialogue with ChatGPT, framing zero-shot IE as multi-turn question-answering. It first identifies possible relation types, and then extracts relational triples based on these types.

2) **GPT-RE** [25] employs a task-aware retrieval model in a few-shot learning framework, incorporating CoT for automatic reasoning, addressing the issues of instance relevance and explanation in input-label mapping.

3) **CodeKGC** [2] uses Python classes to represent structural schemata of relations, enhancing extraction with reasoning rationales.

4) **CodeIE** [10] transforms IE tasks into codes, leveraging LLMs' code reasoning capabilities.

5) **UIE** [16] introduces a structured encoding language for text-to-structured output generation, which is used for pretraining T5 model[19].

6) **USM** [15] proposes a unified semantic matching framework for IE with structured and conceptual abilities, which is built based on RoBERTa [13].

7) **InstructUIE** [27] applies instruction-based fine-tuning on Flan-T5 [5] for enhanced task generalizability.

In brief, ChatIE and CodeKGC utilize zero-shot learning with LLMs, while CodeIE, CodeKGC and GPT-RE adopt few-shot approaches. UIE, USM and InstructUIE adopt supervised fine-tuning (SFT). Notably, GPT-RE was also fine-tuned on larger models like *text-davinci-003* for specific tasks, which is cost-intensive.

4.3 Overall Results and Implementation Details

The overall experimental results are listed in Table 1. Here we only use F1 score as the metric for the alignment with previous papers. For the baseline models, we endeavored to directly cite the scores from their original publications or reproduced the results by using their published models and source codes. Moreover, to ensure the fairness of our experimental comparisons, we predominantly utilized the same backbone LLM, e.g., *gpt-3.5-turbo*. For the methods employing different backbone models, we have included their original results and supplemented them with the results obtained by

---

[3]https://ai.baidu.com/broad/download.
[4]https://nlp.cs.washington.edu/sciIE/.

Table 1: The overall performance (F1 score) of all compared methods on dataset DuIE and SciERC. The best scores and second best scores in each part are bold and underlined, respectively.

Method	Backbone Model	Mode	DuIE	SciERC
ChatIE-single	gpt-3.5-turbo	ZSL	15.61	7.02
ChatIE-multi	gpt-3.5-turbo	ZSL	27.82	12.81
CodeKGC-ZSL	text-davinci-003	ZSL	-	19.90
CodeKGC-ZSL	gpt-3.5-turbo	ZSL	_28.90_	_20.12_
AgentRE-ZSL	gpt-3.5-turbo	ZSL	**32.10**	**23.14**
ChatIE-single	gpt-3.5-turbo	FSL	20.22	8.25
ChatIE-multi	gpt-3.5-turbo	FSL	29.80	11.02
CodeIE	gpt-3.5-turbo	FSL	_32.34_	7.74
CodeKGC-FSL	text-davinci-003	FSL	-	24.00
CodeKGC-FSL	gpt-3.5-turbo	FSL	_35.46_	25.08
GPT-RE	text-davinci-003	FSL	33.42	26.46
GPT-RE	gpt-3.5-turbo	FSL	_35.28_	_26.75_
AgentRE-FSL	gpt-3.5-turbo	FSL	**53.00**	**33.70**
UIE	T5-v1.1-large	SFT	45.72	36.53
USM	RoBERTa-Large	SFT	-	37.36
InstructUIE	11B FlanT5	SFT	_54.32_	45.15
GPT-RE-SFT	text-davinci-003	SFT	-	**69.00**
AgentRE-SFT	Llama-2-7b	SFT	**82.43**	_62.42_

using *gpt-3.5-turbo* as the backbone model, as the italic scores in the table.[5]

Table 1 is divided into three parts based on different experimental paradigms: zero-shot learning (*ZFL*), few-shot learning (*FSL*), and supervised fine-tuning (*SFT*) settings. The second column lists the backbone models used in these methods. For the methods under SFT setting, which can be roughly divided into three categories as below based on the size of the model parameters. 1) The *T5-v1.1-large*[6] used by UIE and the *RoBERTa-Large*[7] used by USM have parameter sizes of 0.77B and 0.35B, respectively. 2) The *Flan-T5*[8] used by InstructUIE and the *Llama-2-7b*[9] used by AgentRE-SFT have parameter sizes of approximately 11B and 7B, respectively. 3) The *gpt-3.5-turbo* used by GPT-RE-SFT has the parameter size of approximately 175B. According to the experimental results, we have the following conclusions.

1. In ZSL group, *ChatIE-multi* outperforms *ChatIE-single*, demonstrating the effectiveness of multi-turn dialogues. *AgentRE-ZSL*'s superior performance indicates its efficient use of auxiliary information.

2. In FSL group, *CodeKGC-FSL* surpasses dialogue-based *ChatIE*, and *GPT-RE* matches its performance, highlighting the benefits of structured reasoning and precise sample retrieval. *AgentRE-FSL* notably outperforms the SOTA models, demonstrating its superior utilization of labelled data and auxiliary information.

---

[5]For CodeKGC, due to its reliance on the now-deprecated *text-davinci-003* model, the replication on DuIE was not feasible. However, we have added the results based on *gpt-3.5-turbo*. Furthermore, for USM and GPT-RE-FT, which necessitate fine-tuning, their non-public model availability precluded the replication on DuIE.

[6]https://github.com/google-research/text-to-text-transfer-transformer.

[7]https://github.com/facebookresearch/fairseq/tree/main/examples/roberta.

[8]https://github.com/facebookresearch/llama.

[9]https://github.com/google-research/FLAN.

Table 2: Ablation study results (Precision, Recall and F1) for the retrieval (R) and memory (M) module on DuIE and SciERC.

Method	DuIE			SciERC		
	Pre.	Rec.	F1	Pre.	Rec.	F1
AgentRE-w/oRM	25.98	32.99	29.04	6.71	8.04	7.48
AgentRE-w/oR	30.75	39.05	34.37	12.19	14.60	13.58
AgentRE-w/oM	38.75	48.41	42.97	19.63	23.52	21.88
AgentRE	**47.42**	**60.21**	**53.00**	**30.23**	**36.21**	**33.70**

Table 3: Ablation study results for different retrieval methods on DuIE and SciERC.

Method	DuIE			SciERC		
	Pre.	Rec.	F1	Pre.	Rec.	F1
None	25.98	32.99	29.04	6.71	8.04	7.48
Random	27.18	33.96	30.14	6.23	7.46	6.94
TF-IDF	31.46	39.30	34.89	15.94	19.09	17.76
BM25	33.10	41.36	36.71	16.77	20.09	18.69
SimCSE	36.88	46.07	40.89	18.68	22.38	20.82
BGE	**38.75**	**48.41**	**42.97**	**19.63**	**23.52**	**21.88**

3. Under SFT setting, fine-tuning smaller models like *UIE* and *USM* yields better results than the baseline models but falls short of *AgentRE-FSL*. *AgentRE-SFT* significantly outperforms *InstructUIE*, evidencing the effectiveness of the distillation learning in AgentRE. However, *GPT-RE-SFT* achieves the best performance on SciERC, albeit with higher training costs due to its large model size and API-based training on *text-davinci-003*.

xi ao

4.4 Ablation and Parameter Tuning Study

This section displays the results of ablation and parameter tuning study, focusing on the impacts of AgentRE's retrieval and memory module on RE performance.

4.4.1 Overall Ablation Study. The ablation study examines the performance of AgentRE under different settings: without the retrieval module (*AgentRE-w/oR*), without the memory module (*AgentRE-w/oM*), and lacking both (*AgentRE-w/oRM*). The results in Table2, reveal a significant underperformance of *AgentRE-w/oRM*, underscoring the essential roles of both modules. *AgentRE-w/oR* and *AgentRE-w/oM* exhibit better performance than *AgentRE-w/oRM*, verifying the value of adding the memory and retrieval module independently. Notably, the full framework AgentRE integrating both modules, achieves the best performance, demonstrating the synergistic effect of combining retrieval capabilities for accessing similar samples and the memory for capitalizing on previous extractions.

4.4.2 Analysis of Retrieval Module. Overall, the variables affecting the retrieval module's effects mainly include the models used for data representation and retrieval and the content available for retrieval.

Retrieval Model: Our experiments evaluated several retrieval methods against the baseline approach, i.e., *Random*, in which *k* labelled samples are chosen at random. These evaluated methods

Table 4: Ablation study results for different retrieval content.

Method	DuIE			SciERC		
	Pre.	Rec.	F1	Pre.	Rec.	F1
None	25.98	32.99	29.04	6.71	8.04	7.48
-samples	29.13	36.38	32.3	14.75	17.68	16.45
-doc	32.84	41.03	36.42	16.64	19.93	18.54
-KG	36.54	45.65	40.52	18.51	22.18	20.64
AgentRE-w/oM	**38.75**	**48.41**	**42.97**	**19.63**	**23.52**	**21.88**

Table 5: Experimental results of two compared methods on DuIE with different amounts of available samples.

#Available sample	AgentRE			ICL		
	Pre.	Rec.	F1	Pre.	Rec.	F1
N=0	33.29	42.27	37.21	20.08	24.37	22.62
N=10	40.64	51.60	45.42	30.08	39.12	34.40
N=100	42.97	53.57	47.75	40.18	51.03	44.91
N=1000	**47.42**	**60.21**	**53.00**	**42.62**	**54.11**	**47.64**

include statistical techniques such as TF-IDF [20] and BM25 [21], as well as embedding-based approaches like SimCSE [6] and BGE [31]. These two schemes employ statistical metrics and vector similarity, respectively, to fetch labelled the samples similar to the given sentence. For implementing TF-IDF and BM25, we utilized the *scikit-learn*[10] and *Rank-BM25*[11] packages, with Chinese word segmentation performed using *Jieba*[12]. The embedding-based models were facilitated through the *SimCSE*[13] package and the *BGE*[14] project. In this set of experiments, the focus was solely on labelled samples, disregarding other relevant information, and the number of retrieved samples was fixed at $k = 5$.

The results in Table 3 demonstrate that both statistical and embedding-based methods significantly surpass the random retrieval baseline. This indicates the effectiveness of retrieving labelled samples more closely aligned with the input text in aiding the model's decision-making process, thereby improving its extraction accuracy. Among the evaluated models, BGE showed superior performance on both datasets and was therefore selected for the retrieval module in subsequent experiments.

Content for Retrieval: Following the backbone model selection for the retrieval module, we delved into the impact of various types of available information for retrieval. As outlined in Section 3.2, this information falls into two main categories: labelled samples and unlabelled relevant information, the latter including annotation guidelines and entity-related KG information.

Table 4 lists the experimental results, where *None* and *AgentRE-w/oM* denote the variants without and only with the full retrieval module, respectively. Additionally, *-samples*, *-doc*, and *-KG* indicate the variants without the labelled sample retrieval, annotation guidelines retrieval, and KG retrieval components, respectively. The results justify that omitting any type of information degrades AgentRE's performance, with the removal of labelled samples (*-samples*) exerting the most significant impact.

In essence, this analysis emphasizes the pivotal roles that both retrieval methodologies and the scope of retrieval content enhance AgentRE's performance. The capabilities of effectively retrieving samples and integrating a broad spectrum of pertinent information are crucial for augmenting AgentRE's extraction proficiency.

4.4.3 Analysis of Memory Module. To evaluate the impact of the memory module on RE performance, we examined the F1, Recall,

and Precision scores of AgentRE with varying memory configurations on the DuIE dataset as training data quantity increased, as depicted in Figure 4 where the X-axis of the figure is the number of training samples. The compared models include *AgentRE-w/oM* (without the memory module), *AgentRE-wM* (with shallow memory as described in Section 3.3.1), and *AgentRE-wM+* (integrating both shallow and deep memory). The models with memory modules leverage both input samples and historical extraction records, unlike their memory-less counterpart. Each model began with an identical set of 200 randomly selected labelled samples for the retrieval module.

The experimental results revealed the following insights:

1) The models incorporating memory module (*AgentRE-wM* and *AgentRE-wM+*) outperform the memory-less variant in all metrics, underscoring the memory module's beneficial impact on extraction accuracy.

2) Performance scores for the models with memory modules improve that as more data was introduced, indicating effective utilization of past extraction experiences for dynamic learning.

3) *AgentRE-wM+* demonstrated superior performance over *AgentRE-wM* with increased data input, suggesting that a comprehensive approach to memory, beyond mere individual sample tracking, can further enhance extraction capabilities.

4.5 Low-Resource Scenario

We also investigated the impact of varying labelled data quantity on extraction performance by sampling different amounts ($N = 0, 10, 100, 1000$) of samples from DuIE. In this study we compared two methods: *AgentRE* integrating retrieval and memory modules, and the basic in-context learning (*ICL*) model employing sample retrieval similar to GPT-RE.

Table 5 lists the relevant results from which we find:

1) The *ICL* model's performance is highly dependent on the quantity of available training samples, with F1 scores of 34.40% and 44.91% at $N = 10$ and $N = 100$, respectively. It highlights the model's limitations in low-resource scenarios, where its dependence on sample retrieval for ICL, without leveraging other pertinent information, adversely affects its extraction capabilities.

2) AgentRE consistently outperforms the ICL model across all data quantities, particularly at extremely low data availabilities ($N = 0, 10$). This suggests AgentRE's superior performance on leveraging the LLM for interaction and reasoning, thus more effectively utilizing available information for enhanced RE.

---

[10]https://scikit-learn.org/stable/
[11]https://github.com/dorianbrown/rank_bm25
[12]https://github.com/fxsjy/jieba
[13]https://github.com/princeton-nlp/SimCSE
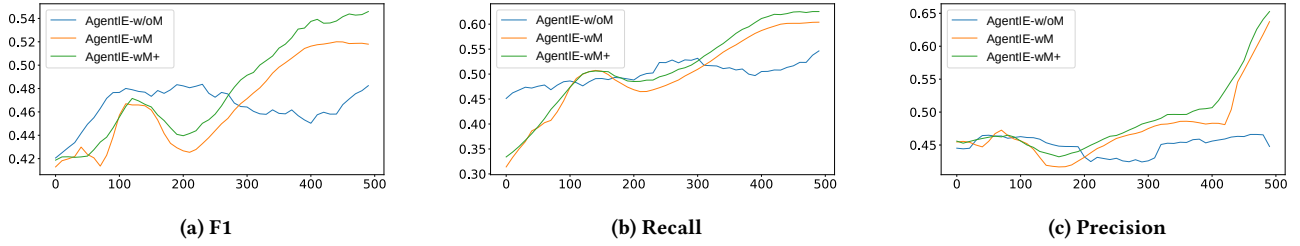[14]https://github.com/FlagOpen/FlagEmbedding

Figure 4: AgentRE's performance on DuIE, including F1, Recall, and Precision with and without the memory module.

Table 6: The performance on DuIE of AgentRE based on two different backbone models with different training data.

Training Data	Llama2-7B			DeepSeek-Coder-7B		
	Pre.	Rec.	F1	Pre.	Rec.	F1
D	79.33	75.19	77.39	80.75	74.91	77.07
D'	81.00	76.78	79.02	81.15	78.92	80.07
$D + D'$	**84.50**	**80.09**	**82.43**	**84.74**	**80.24**	**82.90**

3) Both models exhibit performance gains with increasing N, affirming that additional labelled data promotes model performance by providing more relevant training samples.

4.6 Fine-Tuning Study

In this subsection, we verify the effectiveness of the distillation method based on historical reasoning rationales introduced in Section 3.5. When fine-tuning SLLMs, a straightforward approach is to input the sentence x directly into the model, allowing it to output the predicted triples $\hat{Y}$. The original training data in this manner is denoted as D, and the dataset D' is derived from summarizing the agent's historical reasoning trajectories. By comparing the performance of models trained on these two different datasets, we explore the effectiveness of distillation learning.

Specifically, D includes 10,000 samples from DulE's training set, while D' contains reasoning rationales and 1,000 samples. In addition to comparing the models trained separately on each dataset, we also considered sequential fine-tuning on both datasets, denoted as $D + D'$. This approach involves the initial training on the larger dataset D followed by further fine-tuning on D'. In all experiments, models are trained for 3 epochs on each dataset.

Parameter-efficient fine-tuning was performed using the LoRA[9] method, with the low-rank matrix dimension set to $r = 8$, the scaling factor set to alpha = 16, and the dropout rate set to dropout = 0.1. The optimizer used is AdamW[14], with a learning rate of lr = $5e-5$ and a batch size of bs = 32. For the backbone models, we choose Llama2-7B[23] and DeepSeek-Coder-7B[7]. *Llama2-7B*[15] is one of Meta's general pretrained models with fewer parameters, while *DeepSeek-Coder-7B*[16] is a Chinese and English pretrained model released by DeepSeek AI, pretrained on code and natural language, with a parameter size similar to Llama2-7B.

The experimental results are shown in Table 6, according to which we have the following conclusions.

1) The models fine-tuned on specific training dataset D perform better than the general models trained on multiple datasets (as shown in Table 1), such as UIE, USM, etc. It indicates that targeted fine-tuning for specific extraction tasks can achieve better performance compared to multi-task models.

2) The models fine-tuned on the training dataset D' containing reasoning rationales perform better than those fine-tuned on D, despite the former having significantly less data. It demonstrates that the quality of training data significantly determines the model's performance, and utilizing data derived from the agent's historical reasoning trajectories can better stimulate the reasoning capabilities of smaller models.

3) The experimental results of models trained successively on the two datasets ($D + D'$) reveal that, further fine-tuning on the data with reasoning rationales enhances extraction performance for a model already trained on a large amount of simple labelled data.

5 Conclusion

In this paper, we propose a novel RE framework AgentRE, which effectively leverages various types of information for RE tasks through its retrieval, memory, and extraction modules. The experimental results on two representative datasets demonstrates that our AgentRE achieves satisfactory extraction performance in both zero-shot and few-shot unsupervised learning settings, particularly in low-resource scenarios. Additionally, ablation and parameter tuning studies confirm the significance of each component of AgentRE for the overall extraction performance. Furthermore, AgentRE's reasoning trajectories can form an effective training dataset containing reasoning rationales, facilitating the transfer of capabilities from larger models to smaller models via distillation learning. Due to time and cost constraints, our experiments were conducted on only two representative datasets. Future research will include validating the model on more datasets and extending AgentRE to other information extraction tasks.

Acknowledgments

---

[15]https://github.com/facebookresearch/llama
[16]https://github.com/deepseek-ai/deepseek-coder

References

[1] Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. 2018. Joint entity recognition and relation extraction as a multi-head selection problem. *Expert Systems with Applications* 114 (2018), 34–45.

[2] Zhen Bi, Jing Chen, Yinuo Jiang, Feiyu Xiong, Wei Guo, Huajun Chen, and Ningyu Zhang. 2023. CodeKGC: Code Language Model for Generative Knowledge Graph Construction. In *ACM Transactions on Asian and Low-Resource Language Information Processing*, Vol. abs/2304.09048.

[3] Konstantinos Bousmalis, Giulia Vezzani, Dushyant Rao, Coline Devin, Alex X. Lee, Maria Bauzá, Todor Davchev, Yuxiang Zhou, Agrim Gupta, Akhil S. Raju, Antoine Laurens, Claudio Fantacci, Valentin Dalibard, Martina Zambelli, Murilo Fernandes Martins, Rugile Pevceviciute, Michiel Blokzijl, Misha Denil, Nathan Batchelor, Thomas Lampe, Emilio Parisotto, Konrad Zolna, Scott E. Reed, Sergio Gomez Colmenarejo, Jonathan Scholz, Abbas Abdolmaleki, Oliver Groth, Jean-Baptiste Regli, Oleg O. Sushkov, Tom Rothorl, José Enrique Chen, Yusuf Aytar, David Barker, Joy Ortiz, Martin A. Riedmiller, Jost Tobias Springenberg, Raia Hadsell, Francesco Nori, and Nicolas Manfred Otto Heess. 2023. RoboCat: A Self-Improving Generalist Agent for Robotic Manipulation. https://api.semanticscholar.org/CorpusID:259203978

[4] Baian Chen, Chang Shu, Ehsan Shareghi, Nigel Collier, Karthik Narasimhan, and Shunyu Yao. 2023. FireAct: Toward Language Agent Fine-tuning. *ArXiv* abs/2310.05915 (2023).

[5] Hyung Won Chung, Le Hou, S. Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Wei Yu, Vincent Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed Huai hsin Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling Instruction-Finetuned Language Models. *ArXiv* abs/2210.11416 (2022).

[6] Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *Conference on Empirical Methods in Natural Language Processing*.

[7] Daya Guo, Qihao Zhu, Dejian Yang, Zhenda Xie, Kai Dong, Wentao Zhang, Guanting Chen, Xiao Bi, Yu Wu, Y. K. Li, Fuli Luo, Yingfei Xiong, and Wenfeng Liang. 2024. DeepSeek-Coder: When the Large Language Model Meets Programming – The Rise of Code Intelligence. *arXiv* abs/2401.14196 (2024).

[8] Yucan Guo, Zixuan Li, Xiaolong Jin, Yantao Liu, Yutao Zeng, Wenxuan Liu, Xiang Li, Pan Yang, Long Bai, J. Guo, and Xueqi Cheng. 2023. Retrieval-Augmented Code Generation for Universal Information Extraction. *ArXiv* abs/2311.02962 (2023).

[9] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations (ICLR)*.

[10] Peng Li, Tianxiang Sun, Qiong Tang, Hang Yan, Yuanbin Wu, Xuanjing Huang, Xipeng Qiu Academy for EngineeringTechnology, Fudan University, School of Materials Science, Technology, and East China Normal University. 2023. CodeIE: Large Code Generation Models are Better Few-Shot Information Extractors. *ArXiv* abs/2305.05711 (2023).

[11] Shuangjie Li, Wei He, Yabing Shi, Wenbin Jiang, Haijin Liang, Ye Jiang, Yang Zhang, Yajuan Lyu, and Yong Zhu. 2019. DuIE: A Large-Scale Chinese Dataset for Information Extraction. In *Natural Language Processing and Chinese Computing*.

[12] Xiaoya Li, Fan Yin, Zijun Sun, Xiayu Li, Arianna Yuan, Duo Chai, Mingxin Zhou, and Jiwei Li. 2019. Entity-Relation Extraction as Multi-Turn Question Answering. *ArXiv* abs/1905.05529 (2019).

[13] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[14] Ilya Loshchilov and Frank Hutter. 2017. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.

[15] 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. *Proceedings of the AAAI Conference on Artificial Intelligence* 37, 11 (2023), 13318–13326.

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

[17] Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-Task Identification of Entities, Relations, and Coreference for Scientific Knowledge Graph Construction. In *Conference on Empirical Methods in Natural Language Processing*.

[18] Shuofei Qiao, Ningyu Zhang, Runnan Fang, Yujie Luo, Wangchunshu Zhou, Yuchen Eleanor Jiang, Chengfei Lv, and Huajun Chen. 2024. AUTOACT: Automatic Agent Learning from Scratch via Self-Planning. *ArXiv* abs/2401.05268 (2024). https://api.semanticscholar.org/CorpusID:266902590

[19] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research* 21, 1 (2020), 5485–5551.

[20] Juan Enrique Ramos. 2003. Using TF-IDF to Determine Word Relevance in Document Queries.

[21] Stephen E. Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Inf. Retr.* 3 (2009), 333–389.

[22] Theodore R Sumers, Shunyu Yao, Karthik Narasimhan, and Thomas L Griffiths. 2023. Cognitive architectures for language agents. *arXiv* abs/2309.02427 (2023).

[23] Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, A. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. *ArXiv* abs/2307.09288 (2023).

[24] Dennis Ulmer, Elman Mansimov, Kaixiang Lin, Justin Sun, Xibin Gao, and Yi Zhang. 2024. Bootstrapping LLM-based Task-Oriented Dialogue Agents via Self-Talk. *ArXiv* abs/2401.05033 (2024). https://api.semanticscholar.org/CorpusID:266902624

[25] 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* abs/2305.02105 (2023).

[26] Lei Wang, Chengbang Ma, Xueyang Feng, Zeyu Zhang, Hao-ran Yang, Jingsen Zhang, Zhi-Yang Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Ji-rong Wen. 2023. A Survey on Large Language Model based Autonomous Agents. *arXiv* abs/2308.11432 (2023).

[27] Xiao Wang, Wei Zhou, Can Zu, Han Xia, Tianze Chen, Yuan Zhang, Rui Zheng, Junjie Ye, Qi Zhang, Tao Gui, Jihua Kang, J. Yang, Siyuan Li, and Chunsai Du. 2023. InstructUIE: Multi-task Instruction Tuning for Unified Information Extraction. *CoRR* abs/2304.08085 (2023).

[28] Yucheng Wang, Bowen Yu, Yueyang Zhang, Tingwen Liu, Hongsong Zhu, and Limin Sun. 2020. TPLinker: Single-stage joint extraction of entities and relations through token pair linking. *arXiv preprint arXiv:2010.13415* (2020).

[29] Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, Yong Jiang, and Wenjuan Han. 2023. Zero-Shot Information Extraction via Chatting with ChatGPT. *arXiv* abs/2302.10205 (2023).

[30] Likang Wu, Zhilan Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, Hui Xiong, and Enhong Chen. 2023. A Survey on Large Language Models for Recommendation. *CoRR* abs/2305.19860 (2023).

[31] Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-Pack: Packaged Resources To Advance General Chinese Embedding. *ArXiv* abs/2309.07597 (2023).

[32] Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, and Enhong Chen. 2023. Large Language Models for Generative Information Extraction: A Survey. *arXiv* abs/2312.17617 (2023).

[33] Shunyu Yao, Jeffrey Zhao, Dian Yu, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2022. ReAct: Synergizing Reasoning and Acting in Language Models. In *NeurIPS 2022 Foundation Models for Decision Making Workshop*.

[34] Hongbin Ye, Honghao Gui, Aijia Zhang, Tong Liu, Wei Hua, and Weiqiang Jia. 2023. Beyond Isolation: Multi-Agent Synergy for Improving Knowledge Graph Construction. *ArXiv* abs/2312.03022 (2023). https://api.semanticscholar.org/CorpusID:265696391

[35] Bowen Yu, Zhenyu Zhang, Xiaobo Shu, Yubin Wang, Tingwen Liu, Bin Wang, and Sujian Li. 2019. Joint extraction of entities and relations based on a novel decomposition strategy. *arXiv preprint arXiv:1909.04273* (2019).

[36] Kai Zhang, Bernal Jimenez Gutierrez, and Yu Su. 2023. Aligning Instruction Tasks Unlocks Large Language Models as Zero-Shot Relation Extractors. In *Findings of the Association for Computational Linguistics: ACL 2023*. Association for Computational Linguistics, 794–812.

[37] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Z. Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jianyun Nie, and Ji rong Wen. 2023. A Survey of Large Language Models. *ArXiv* abs/2303.18223 (2023).

[38] Shaowen Zhou, Yu Bowen, Aixin Sun, Cheng Long, Jingyang Li, Haiyang Yu, and Jianguo Sun. 2022. A Survey on Neural Open Information Extraction: Current Status and Future Directions. *ArXiv* abs/2205.11725 (2022).

[39] Wenxuan Zhou, Sheng Zhang, Yu Gu, Muhao Chen, and Hoifung Poon. 2024. UniversalNER: Targeted Distillation from Large Language Models for Open Named Entity Recognition. In *The Twelfth International Conference on Learning Representations*, Vol. abs/2308.03279.

[40] Yucheng Zhou, Tao Shen, Xiubo Geng, Guodong Long, and Daxin Jiang. 2022. ClarET: Pre-training a Correlation-Aware Context-To-Event Transformer for Event-Centric Generation and Classification. *ArXiv* abs/2203.02225 (2022).

[41] Yuqi Zhu, Shuofei Qiao, Yixin Ou, Shumin Deng, Ningyu Zhang, Shiwei Lyu, Yue Shen, Lei Liang, Jinjie Gu, and Huajun Chen. 2024. KnowAgent: Knowledge-Augmented Planning for LLM-Based Agents. *ArXiv* abs/2403.03101 (2024). https://api.semanticscholar.org/CorpusID:268248897

[42] Yuqi Zhu, Xiaohan Wang, Jing Chen, Shuofei Qiao, Yixin Ou, Yunzhi Yao, Shumin Deng, Huajun Chen, and Ningyu Zhang. 2023. LLMs for Knowledge Graph Construction and Reasoning: Recent Capabilities and Future Opportunities. *ArXiv* abs/2305.13168 (2023).