Leveraging LLMs Few-shot Learning to Improve Instruction-driven Knowledge Graph Construction

Yongli Mou
RWTH Aachen University
Aachen, Germany
mou@dbis.rwth-aachen.de

Li Liu
RWTH Aachen University
Aachen, Germany
li.liu@rwth-aachen.de

Sulayman Sowe
RWTH Aachen University
Aachen, Germany
Fraunhofer FIT
Sankt Augustin, Germany
sowe@dbis.rwth-aachen.de

Diego Collarana
Fraunhofer FIT
Sankt Augustin, Germany
Universidad Privada Boliviana
Cochabamba, Bolivia
diego.collarana.vargas@fit.fraunhofer.de

Stefan Decker
RWTH Aachen University
Aachen, Germany
Fraunhofer FIT
Sankt Augustin, Germany
decker@dbis.rwth-aachen.de

ABSTRACT

Large Language Models (LLMs) continue to demonstrate immense capabilities in many tasks relating to natural language understanding and generation. These capabilities, to a large extent, are possible because of the effective data management strategies that go into preparing the training dataset for these models. Applying a few-shot learning technique on a dataset opens unique opportunities for using LLMs in many domains, including enhancing knowledge graph construction (KGC) processes. However, the fundamental problem in KGC is identifying entities and relationships and resolving triple complexities. In this work, we explore the in-context learning capability of GPT-4 for instruction-driven adaptive KGC and propose a novel approach that forces GPT-4 to reflect on the errors it makes in the given examples and generates verbal experience to guide the model to avoid similar mistakes during the KGC. Our comparative analysis of few-shot learning and the baseline (zero-shot learning) not only highlights the strengths and limitations of GPT-4 in KGC but also demonstrates how the in-context learning capabilities of GPT-4 can contribute to more dynamic, accurate, and instruction-followed knowledge graphs.

VLDB Workshop Reference Format:
Yongli Mou, Li Liu, Sulayman Sowe, Diego Collarana, and Stefan Decker. Leveraging LLMs Few-shot Learning to Improve Instruction-driven Knowledge Graph Construction. VLDB 2024 Workshop: LLM+KG.

VLDB Workshop Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://git.dbis.rwth-aachen.de/code/llmkg-workshop-vldb-2024.

Proceedings of the VLDB Endowment. ISSN 2150-8097.

1 INTRODUCTION

Human knowledge representation, learning and reasoning is a fundamental cornerstone of Artificial Intelligence (AI) [14]. Knowledge Graphs (KGs) are designed to accumulate and convey factual knowledge of the real world, which organize and represent knowledge in the form of graphs, where the nodes represent entities of interest or attributes and edges represent relations between these entities and attributes [22]. As a form of structured human knowledge, KGs greatly complement the domain of AI by providing machine-readable data and semantic richness, which allows for more sophisticated and context-aware algorithms. KGs have been successfully applied in various areas, including information retrieval [18], recommendation systems [38] and question answering [4].

Constructing KGs involves a systematic process that extracts information from various sources. Early approaches for KG construction were predominantly human-centric, i.e., extensively relying on human expertise and involving considerable manual curation, domain-specific knowledge, and rule-based extraction techniques to identify and categorize entities and their interrelationships. Representative examples of these early methods include TextRunner [45], KnowItAll [28], Yago [31] and DBpedia [3]. Although those approaches ensure a high degree of reliability in the resulting KGs, they are laborious and scale poorly with the vast amount of information available on the web. Additionally, while powerful in specific domains, rule-based systems struggle to adapt to the dynamic and varied nature of human knowledge, limiting their applicability and flexibility. Subsequently, several machine learning-based approaches have been proposed for specific tasks in the KG construction process, such as named entity recognition [9, 48], coreference resolution [30], and relation extraction [46]. Leveraging those machine learning-based methods to build a pipeline that systematically addresses individual tasks can automate and scale the construction of KGs. Despite the significant benefits, a notable concern is the issue of error propagation, where inaccuracies in early processing stages can cumulatively affect the quality and reliability of the resulting knowledge graph.

Large Language Models (LLMs), such as the OpenAI GPT series, have recently demonstrated remarkable language understanding and text generation capabilities. One of the most remarkable aspects of LLMs, including GPT-4, is their ability to perform in-context learning (or so-called few-shot learning), which enables LLMs to adapt their responses based on the examples provided within a prompt [5]. In this paper, we aim to explore the potential of GPT-4 for KG construction driven by user instructions. A critical aspect of our exploration focuses on the in-context learning capabilities of GPT-4, with which their responses can be adapted based on the examples provided in the prompt, making selecting these examples a pivotal factor in the effectiveness of the KG construction. We test different example selection strategies, examining their impacts on the quality of generated KGs. Moreover, we propose a self-reflection approach that enables GPT-4 to explicitly or implicitly reflect on errors from examples to mitigate incorrect inference during the construction process, which involves instructing the LLM to identify and learn from the mistakes in its outputs or the examples provided, thereby enhancing its ability to generate more accurate and reliable KGs.

2 BACKGROUND AND RELATED WORK

In this section, we provide a comprehensive overview of the foundational concepts and significant advancements in LLMs and KG construction, which sets the stage for understanding how these technologies can be integrated to enhance instruction-driven KG construction.

2.1 Large Language Models

Recently, we have witnessed the great success of LLMs, which have shown great capabilities in understanding and generating human language and led to breakthroughs in a wide range of downstream applications, including text summarization, machine translation, content generation, and conversational systems.

Architectural Foundations. Modern LLMs are mostly built upon Transformer architecture [35], known for its self-attention mechanism. The transformer architecture has been adapted into three main forms: Encoder-decoder such as T5 [24], Encoder-only such as BERT [7] and Decoder-only such as GPT [23]. Nowadays, the training paradigm for LLMs typically includes two main phases, i.e., first pre-trained on a massive and diverse dataset and then fine-tuned on a smaller and domain-specific dataset allowing for specialized performance in specific domains or tasks. We have witnessed the great success of commercial LLMs, such as OpenAI GPT family and Anthropic Claude family. In addition to proprietary models, the field of large language models (LLMs) has seen the development of several influential open-source models. Notable examples include ALBERT [15], which reduces model size while maintaining performance through parameter sharing and factorized embedding parameterization. RoBERTa [16] builds on BERT by optimizing the training approach, such as training with larger mini-batches and removing the next-sentence prediction task. DistilBERT [26] focuses on model compression, providing a smaller, faster version of BERT that retains most of its performance. More recent models like LLaMa [32, 33] and Bloom [27] push the boundaries of open-source LLMs by offering competitive performance with state-of-the-art proprietary models.

Prompting Techniques and In-Context Learning. Hallucination in the context of LLMs refers to the generation of information that is not supported by the input data or existing knowledge. This phenomenon can lead to factually incorrect or misleading responses, undermining the model's utility in critical applications. Prompting techniques in the context of language models refer to crafting inputs (prompts) to guide LLMs in generating specific outputs, which is crucial for extracting maximum utility from these models. Several prompting techniques have been proposed, such as Chain-of-Thought (CoT), Chain-of-Thought with Self-Consistency (CoT-SC), ReAct and Reflexion. CoT [40] encourages models to think aloud by generating intermediate reasoning steps before arriving at a final answer. This approach not only improves the interpretability of model outputs but also enhances the accuracy of complex reasoning tasks by breaking them down into simpler and manageable steps. Building on it, CoT-SC [39] incorporates self-consistency mechanisms that generate multiple reasoning paths for a given task and select the most consistent answer among them, thereby improving the reliability and accuracy of the model outputs. ReAct [44] leverage LLMs to generate reasoning traces and task-specific actions in an interleaved manner, where reasoning traces create and update plans and handle exceptions and actions allow to interact with external sources to obtain further information. Reflexion [29] reinforces language models through linguistic feedback and maintains those reflective texts in an episodic memory buffer for better decision-making in subsequent trials.

Besides, in-context learning is a significant stride in LLMs, which enables LLMs to adapt responses based on the context provided within a prompt and to understand and execute tasks with limited examples. In-context learning is a paradigm that allows language models to learn tasks given only a few examples in the form of demonstration [8]. GPT-3 has demonstrated substantial proficiency in performing a wide array of NLP tasks with minimal examples. Brown et al. tested GPT-3 across various settings: zero-shot, one-shot, and few-shot settings and the model's performance achieved state-of-the-art results in many cases without the need for task-specific fine-tuning, which showcased its capabilities in few-shot learning [5]. Building on the success of monolingual models, recent studies have explored few-shot learning in multilingual contexts. Research has shown that multilingual generative language models, such as those trained on diverse language corpora, can effectively perform few-shot learning across different languages [17]. Several efforts have been made to improve the efficiency and effectiveness of few-shot learning. Gao et al. [10] introduced a computationally efficient prompt-based fine-tuning approach with automated prompt generation and a refined strategy for dynamically and selectively incorporating demonstrations into contexts.

2.2 Knowledge Graph Construction

Definition of Knowledge Graphs. KGs represent knowledge in the real world in the form of graphs, whose nodes represent entities of interest or attributes and whose edges represent relations between these entities and attributes. Formally, a KG is defined as $\mathcal{K} = (\mathcal{E}, \mathcal{R}, \mathcal{T}, \mathcal{F})$ consisting of a set of entities and attributes $\mathcal{E}$, a set of relations $\mathcal{R}$, a set of factual triples $\mathcal{T} = \{(h, r, t)\}$ where $h, t \in \mathcal{E}$ and $r \in \mathcal{R}$ (typically binary fact), $\mathcal{F}_k$ a set function representing the background knowledge, or in practice, a rule set, schema, or set of implicit math principles, that constrains potential facts to be knowledge-level informative $\mathcal{T} \subset \mathcal{F}_k(\{\mathcal{E}, \mathcal{R}\})$ [47].

Definition of Knowledge Graph Construction. The construction of KGs has been an active research area in recent years, with various approaches proposed to build and enhance KGs. Constructing KGs is a task of extracting information from various data sources (structured, semi-structured, and unstructured data) and storing it in a graph structure. The general process of KG construction can be divided into three main steps, including knowledge acquisition, knowledge refinement and knowledge evolution. Entity Discovery and Relation Extraction (RE) are foundational tasks in the phase of knowledge acquisition [47]. Named Entity Recognition (NER) and Entity Linking (EL) are key subtasks of Entity Discovery. NER aims to locate (entity detection) and classify (entity typing) named entities mentioned in unstructured text into predefined categories such as the names of persons, organizations, locations, etc [14]. Entity Typing (ET) is sometimes treated as a distinct task of NER, as a unique challenge. Entity Linking (EL) or entity disambiguation is a unified task that links recognized entities to the corresponding entities in the existing knowledge base (for example, standard ontologies) [14]. The knowledge acquisition process is complemented by Entity Typing (ET), Entity Linking (EL) and Coreference Resolution (CR), each contributing uniquely to the enrichment and precision of information extraction and KG construction. These tasks, NER, EL, CR and RE, form a comprehensive framework for developing pipelines that transform unstructured data into structured, interconnected data, i.e., KGs. Traditionally, these pipelines heavily relied on linguistic approaches to solve the underlying task, albeit constrained by their inherent limitations. With the recent emergence of LLMs, numerous studies have utilized these models for NER and RE, achieving promising results. Additionally, many end-to-end approaches have also been developed.

Large Language Models-based Knowledge Graph Construction. With the success of ChatGPT, LLMs have been studied to applied to the construction of KGs. Chen et al. [11] compare the ability of in-context learning of GPT-3 under the few-shot setting with smaller BERT-size models on biomedical information extraction tasks, including NER and RE. GPT-NER [37] transforms NER from a sequence labeling task to a generation task that can be dealt by LLMs by marking entities with tokens "@@##" based on GPT-3. It then employs a self-verification strategy, prompting LLMs to assess whether the extracted entities match the labeled entity tags. VicunaNER [13] is a two-phase zero/few-shot NER framework based on Vicuna, an open-source LLM, which leverages multi-turn dialogues to recognize entities from texts. It involves a Recognition phase to identify and verify entities and a Re-Recognition phase to find and correct any missed entities. PromptNER [2] requires additional entity definitions beyond standard few-shot examples to implement few-shot and cross-domain NER with T5-Flan-11B, GPT-3.5 and GPT-4. It generates a list of potential entities from a sentence, with explanations confirming their alignment with specified entity types.

Wadhwa et al. [36] evaluate GPT-3 and Flan-T5 (large) on standard RE tasks under different levels of supervision. Its performance can be significantly enhanced through supervision and fine-tuning using CoT-style explanations. Xu et al. [43] adopts in-context learning and data generation on few-shot RE via GPT-3.5 and proposes task-related instructions and schema-constrained data generation to enhance the performance under the few-shot setting. S2ynRE [42] is a framework of two-stage Self-training with Synthetic data for RE based on GPT-2-Large and GPT-3.5. It first generates synthetic training data by adapting the LLM to the target domain, then iteratively learns from both synthetic and gold data with a self-training algorithm. Chen et al. [1] implement a student model for multimodal NER and RE by generating CoT prompts based on BERT-base-uncased, XLM-RoBERTa-large, GPT-3.5-turbo and GPT-4. They propose a conditional prompt distillation method to absorb the commonsense reasoning ability of LLMs.

Besides, there are also integrated methods that deal with NER and RE together. AERJE (API Entity-Relation Joint Extraction framework) [12] is a T5-based model extracting API entities and relations from unstructured text. It uses a multi-task architecture and dynamic prompts to frame the extraction as a sequence-to-sequence generation task. Trajanoska et al. [34] conduct experiments with sustainability-related text to compare foundational LLMs like ChatGPT with specialized pre-trained models like REBEL for joint extraction in a pipeline-based method. ChatIE [41] converts zero-shot IE tasks into multi-turn question answering problems with a two-stage framework based on ChatGPT and GPT-3. It first identifies possible element types in the text and then performs a chain-styled IE on each identified type.

3 METHODS

In this section, we describe the methods proposed for integrating LLMs into the process of KG construction and mainly focus on the task of instruction-driven adaptive KG construction. We adopt end-to-end approaches, i.e., LLMs directly generate the output in the form of triples. Our approach primarily leverages the in-context learning capabilities of LLMs to generate knowledge graph triples directly from user inputs and instructions.

3.1 Task Definition

The task instruction-driven adaptive KG construction (InstructionKGC) was first defined in the CCKS2023 Competition Task 1 that is extracting entities and relations of specific types based on the input text and the instructions provided by the user to construct a KG[1]. The core objective is to update and optimize the representation of the KG based on user requirements, thereby achieving more accurate and efficient information retrieval and reasoning [6], and meeting the demands for efficient KG construction in open environments. In this work, there are assumptions for testing our proposed methods based on the task definition and data provided by the CCKS2023 Competition (Details in Section 4.1.1). First, we assume that we have a pool of labeled data as examples for few-shot learning. Second, we assume that all data consists of features including ID, category, input and instruction (labeled data with output in the form of triples).

[1]https://tianchi.aliyun.com/competition/entrance/532080

3.2 Few-shot Learning and Example Selection Strategies

Our approach primarily leverages the in-context learning capabilities of LLMs, which enables the LLM to generate output with a small number of examples. As mentioned before, we assume a labelled data pool of instruction-input-output pairs, from which examples are selected. Due to the context window, the phenomenon “lost in the middle” is observed in retrieval augmented generation approaches [20]. In our case, we set the number of examples (or "shots") as 3 and applied 3-shot learning to design demonstrations. Initially, several examples are selected and converted into demonstrations with defined formats. These demonstrations are then concatenated with a query question to build a prompt. The prompt we use for naïve 3-shot learning is shown in Figure 1.

Naïve few-shot learning prompt

```
# Example 1:
Instruction: {$instruction_1}
Input: {$input_1}
Output: {$output_1}
# Example 2:
Instruction: {$instruction_2}
...
# Question:
Instruction: {$instruction_q}
Input: {$input_q}
Output: model generated output in the form of a list of
triples, same as example outputs [(h1, r1, t1), ...]
```

Figure 1: Naïve few-shot learning prompt template used for instruction-driven knowledge graph construction.

The effectiveness of few-shot learning relies heavily on the examples provided during the instruction phase as these examples guide the LLMs in understanding the context and structure of the desired output [19, 21]. The selection strategy can be grounded in three different criteria such as relevance, representativeness, and diversity. Relevance ensures that the examples are directly related to the task at hand, facilitating a more focused in-context learning process. Representativeness ensures that the examples reflect the wide range of possible inputs and desired outputs, covering various types of entities, relationships, and domain-specific nuances. Diversity aims at presenting a broad spectrum of scenarios, preventing the model from overfitting to specific patterns and promoting its ability to generalize across different knowledge domains. We implement and compare 4 example selection strategies to maximize the model performance and get the best results, including fixed examples, random selection, category-based selection, and similarity-based selection.

Fixed examples. In order to compare with other strategies and show the necessity of the example selection, three examples are chosen from the same category as the most basic strategy to build demonstrations.

Random Selection. Since there are 12 categories in total in the dataset, we randomly choose 3 examples from the whole example dataset. These examples could be chosen from different categories and this increases the randomness of demonstrations.

Category-based Selection. Relation lists and input text could be disparate or similar depending on categories. So we also randomly choose 3 examples per category to eliminate the negative impacts from different categories with incomparable text input.

Similarity-based Selection. The examples are chosen to explicitly demonstrate how entities and relations should be extracted from texts and instructions similar to those models will encounter in real scenarios and, therefore, must be closely related to the instruction and text input. We implement similarity-based selection in two phases,i.e., embedding and retrieval (or search). For simplicity, we embed the example instruction-text pairs and the query by using Sentence-BERT [25] with paraphrase-MiniLM-L12-v2 model [2] and calculate the cosine similarities.

3.3 Self-Reflection

Similar to Reflexion [29], we propose the self-reflection mechanism, designed to enable LLMs to critically evaluate their output and learn from errors using examples. This process not only aids in error identification and correction but also in enhancing the capability of LLMs to generate more accurate and reliable KG triples in future iterations. Figure 2 shows the overview of our proposed self-reflection based in-context learning approach for InstructKGC.

After the selection of examples, the LLM predicts outputs based on the instructions and input texts. Then we instruct the LLM to review its outputs and identify potential errors or inconsistencies, which involves the comparison of generated entities and relations against the golden output of the examples. Once errors are identified, the LLM is guided to reflect on these mistakes and generate verbal experiences. Finally, the query, the selected examples and the experiences summarized are formed as the prompt to instruct LLM to adjust its internal representations and strategies for generating KG triples, aiming to avoid similar errors in future responses. Figure 3 and 4 show prompts used for self-reflection based few-shot learning, where the first one implicitly summarizes the mistakes made in the predictions of example outputs while the second one explicitly generate verbal error summarization.

4 EVALUATION

In this section, we detail the evaluation used to assess the efficacy of integrating LLMs into InstructionKGC, particularly our proposed self-reflection based in-context learning approach.

4.1 Experimental Settings

4.1.1 Dataset. We conduct our experiments on the dataset from CCKS2023 competition Task 1, i.e., the instruction-driven adaptive KG construction task. An example data is shown in Listing 1. The competition dataset consists of three subsets for training, validation

[2] https://huggingface.co/sentence-transformers/paraphrase-MiniLM-L12-v2

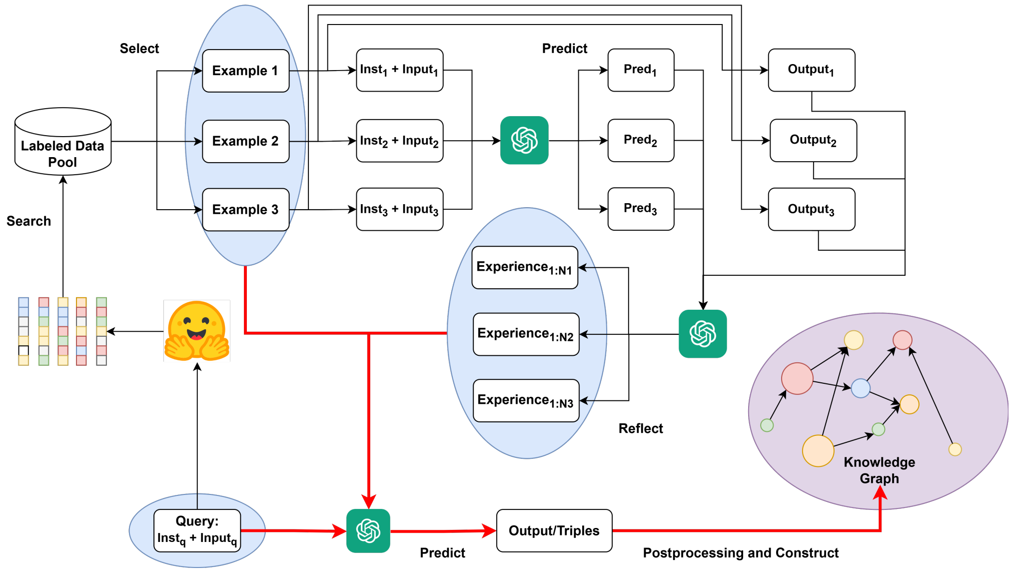

Figure 2: Overview of the self-reflection-based in-context (few-shot) learning for InstructKGC. Parts in light blue ellipses form the self-reflected prompt and the light purple ellipse represents output.

and testing. However, only the training part is provided with labels and so far there is no API available for continuous evaluation for validation and testing. Besides, our proposed approaches do not involve fine-tuning currently. Therefore we only use the training part provided by the competition including 5000 data items with instruction, input and output for our evaluation. There are 12 categories in total, including building, transport, astronomy, organization, etc. And different categories may contain the same relationships. Due to the high cost of calling the LLM API, we randomly select 1000 data for our experiments for testing. To avoid the negative impact of the possible distinct quantity of different categories, 90 examples were randomly and uniformly selected per category from the rest to build the example pool for in-context learning.

4.1.2 Metrics. The performance of our methods is evaluated using graph-level and entity-level metrics to provide a holistic view of the effectiveness of LLMs for KGC. We employ the metrics of precision, recall, F1-score and ROUGE-2 to quantify the performance on graph level, while precision, recall and F1-score for entity-level evaluation.

It is worth noting that the graph-level metric ROUGE-2 we used is different from the one defined in the competition. During the whole evaluation process, we treat entity as a basic unit, which means that exact match is conducted on both entity and tuple-level. And the evaluation metric adopted in the competition is treating the whole output as a string. We debate that this approach is not as meaningful as ours because there may be unreasonable combinations of two neighboring triples. Thus, our results are not comparable with other results from the competition. In order to gauge the similarity between the predicted knowledge and the golden knowledge, we utilize the overlap of N-grams. This involves converting the predicted knowledge and the golden knowledge with triple format into pairs of two elements, i.e. convert the triples (h, r, t) into two pairs (h, r) and (r, t). Subsequently, these pairs are employed in the computation of ROUGE-N [16], with N set to 2.

4.1.3 Implementation Details. Experiments are conducted using GPT-4 model from OpenAI[3]. It is a large multilingual model that accepts text or image inputs and emits text outputs and the longest context can be up to 8,192 tokens. For the model parameters, we set the maximum output length to 8,192 tokens. Temperature is set to 1.0, and others to the defaults, for example, top-p to 1.

We choose the method of zero-shot learning as the baseline, which directly prompts with instructions and input into the model and generates outputs. This corresponds to the base method of the baseline EasyInstruct[4]provided by CCKS2023.

Before the methods are implemented, the training dataset from the competition is split into an example pool set and a predicting set, and similarities between examples and predicting text are calculated beforehand for ease of later implementation. During the experiments, different fields from examples and additional information are concatenated and fed into the LLM, generating text strings as outputs. Subsequently, the generated outputs are formalized with regular expressions to transform text strings into ideal entity relation triples.

4.2 Results

Our experimental results of the graph-level evaluation and entity-level evaluation are shown in Table 1 and 2 respectively. In our research, we compare the end-to-end approaches in the following settings: zero-shot learning as the baseline, zero-shot learning with CoT, few-shot learning (we set to 3-shot learning) with different example selection strategies and few-shot learning with implicit and

[3]https://platform.openai.com/docs/models/gpt-4-turbo-and-gpt-4
[4]https://github.com/zjunlp/EasyInstruct

Few-shot learning prompt with implicit self-reflection

Example 1:
Instruction:
Given a list of candidate relations: ['Alias', 'Type'], please extract possible subject and object entities the following input based on the relation list and provide the corresponding relationship triples. Please respond in the format (Subject, Relation, Object).
Input:
Tianyin Zengsan (i.e. 13Tau) is an augmenting star of the Pleiades in the ancient Chinese system of astrological officials, and is located in the constellation of Taurus according to the modern classification of Western constellations. It has a surface temperature of 11,000-25,000 K, a colour index of -0.016, and an absolute magnitude of about 0.187.
Output:
The correct output triples should be: (Tianyin Zeng San, Alias, 13Tau), (Xingguan, Type, constellation), (Tianyin Xingguan, Type, constellation), (Pleiades, Type, constellation), (Taurus, Type, constellation)
Example 2:
Instruction:
...
Compare the generated triples with the actual triples and summarize the errors, avoiding them in the following tasks.
Question:
Instruction:
Given a list of candidate relations: ['Alias', 'Belongs to', 'Type"], please extract possible subject and object entities the following input based on the relation list and provide the corresponding relationship triples. Please respond in the format (Subject, Relation, Object).
Input:
79b Cetus (also known as HD 16141b) is an exoplanet in the constellation Cetus with an orbital period of 75 Earth days around 79 Cetus. It and HD 46375b were both discovered on 29 March 2000, and were the first exoplanets discovered with lower mass limits smaller than Saturn.
Output:
(79b Cetus, Alias, HD16141b), (79b Cetus, Belongs to, Cetus), (79b Cetus, Type, exoplanet), (HD46375b, Belongs to, Cetus), (HD46375b, Type, exoplanet)

Figure 3: Few-shot learning prompt template with implicit self-reflection used for instruction-driven knowledge graph construction (Purple colored text from examples, green colored text from GPT-4's output).

Methods		Selection	Precision	Recall	F1-Score	ROUGE-2
End-to-end	Zero-shot Learning	No	27.28	17.49	21.31	28.08
	Zero-shot Learning (CoT)	No	24.00	17.46	20.21	30.54
	Few-shot Learning	Fixed	31.33	24.29	27.37	39.24
		Random	30.67	24.72	27.37	40.05
		Category	37.93	33.55	35.61	49.35
		Similarity	31.19	25.33	27.96	40.16
	Few-shot Learning (SR_i)	Category	41.01	33.86	37.10	47.79
	Few-shot Learning (SR_e)	Category	39.58	34.88	37.08	49.55
Pipeline-based	Zero-shot Learning	No	23.29	16.39	19.24	28.68

Table 1: Graph-level evaluation results. CoT: Chain-of-Thought, SR_i: Implicit self-reflection, SR_e: Explicit self-reflection.

Few-shot learning prompt with explicit self-reflection

Example 1:
Instruction:
Given a list of candidate relations: ['Discoverer', 'Origin of name'], please extract possible subject and object entities the following input based on the relation list and provide the corresponding relationship triples. Please respond in the format (Subject, Relation, Object).
Input:
The Italian astronomer Galileo Galilei discovered Io in January 1610 along with three other large moons of Jupiter (Io, Io and Io). Io takes its name from Callisto, one of Zeus' lovers in Greek mythology, a nymph (sometimes thought to be the daughter of Lycaon) who was close to the moon goddess Artemis. Simon Marius suggested the name shortly after the star was discovered, and Marius attributed it to Johannes Kepler. However astronomers did not welcome the name for a long time, and it was not widely adopted until the mid-20th century. Many early astronomical texts refer to the moon by its Roman numeral (a system proposed by Galileo), i.e. as Jupiter IV or "the fourth satellite of Jupiter".
The correct triples are:
(Io, Discoverer, Galileo), (Io, Discoverer, Galileo), (Io, Discoverer, Galileo), (Io, Discoverer, Simon Marius), (Io, Discoverer, Galileo), (Io, Discoverer, Galileo), (Jupiter, Origin of name, Jupiter)
The error generated by the model is: The output of the model is correct for the four moons discovered by Galileo (Io, Io, Io and Io), but the model does not provide the correct output for Simon Marius, the discoverer of Io. Also, the model does not give the correct triad that the origin of Jupiter's name is Jupiter. In the output ternary, the model incorrectly gives (Io, 'origin of the name', 'Callisto'), which is wrong, Io was not named by 'Callisto'. In summary, the model has a correct understanding of some basic astronomical knowledge, but still has some errors when dealing with specific knowledge facts. The model may need further training to improve its accuracy in processing specific knowledge facts. Avoid these errors in the next tasks.
Example 2:
Instruction:
...
Avoid these errors in the following tasks.
Question:
Instruction:
Given a list of candidate relations: ['Alias', 'Belongs to', 'Type"], please extract possible subject and object entities the following input based on the relation list and provide the corresponding relationship triples. Please respond in the format (Subject, Relation, Object).
Input:
79b Cetus (also known as HD 16141b) is an exoplanet in the constellation Cetus with an orbital period of 75 Earth days around 79 Cetus. It and HD 46375b were both discovered on 29 March 2000, and were the first exoplanets discovered with lower mass limits smaller than Saturn.
Output:
(79b Cetus, Alias, HD16141b), (79b Cetus, Belongs to, Cetus), (79b Cetus, Type, exoplanet), (HD46375b, Type, exoplanet)

Figure 4: Few-shot learning prompt template with explicit self-reflection used for instruction-driven knowledge graph construction (Purple colored text from examples, green colored text from GPT-4's output).

explicit self-reflection. Additionally, we also implement a pipeline-based approach that splits the KG construction process into named entity recognition, coreference resolution, and relation extraction.

As our baseline, in the zero-shot learning, where no specific examples were provided to the model beforehand, we observed a

```
{
  "id": "0",
  "cate": ["transport", "instruction"],
  "instruction": "A list of known candidate relationships, ['Creation time',
                 'Administrative territory', 'Station class', 'Route', 'Postal code'],
                 please extract the possible head entity (Subject) and tail entity
                 (Object) from the following inputs, based on the list of relationships,
                 and give the corresponding relationship triples. Please answer in the
                 format (Subject,Relation,Object).",
  "input": "Shaheji Station is a railway station on the Beijing-Shanghai line, located
             in Longting Community, Shahe Town, Nanqiao District, Chuzhou City,
             Anhui Province, built in 1909, currently a ..." ,
  "output": "(Shaheji Station, line, Beijing-Shanghai Line),
            (Shaheji Station, administrative territory, Longting Community),
            ...
            (Shaheji Station, postal code, 239060),
            (Beijing-Shanghai line, administrative territory, Longting community)",
  "kg": [["Shaheji Station", "line", "Beijing-Shanghai Line"], ["..."]]
}
```

Listing 1: Example data from CCKS 2023 competition Task 1

Approaches	Methods	Selection	Precision	Recall	F1-Score
End-to-end	Zero-shot Learning	No	57.89	48.61	52.85
	Zero-shot Learning (CoT)	No	52.17	46.11	48.95
	Few-shot learning	Fixed	64.03	59.68	61.78
		Random	62.40	59.73	61.04
		Category	66.44	67.76	67.10
		Similarity	62.29	60.19	61.22
	Few-shot Learning (SR_i)	Category	67.14	64.57	65.83
	Few-shot Learning (SR_e)	Category	65.89	65.36	65.62
Pipeline-based	Zero-shot Learning	No	56.33	50.73	53.39

Table 2: Entity-level evaluation results in percentage. CoT: Chain-of-Thought, SR_i: Implicit self-reflection, SR_e: Explicit self-reflection.

precision of 27.28%, a recall of 17.49%, an F1-score of 21.31%, and a ROUGE-2 score of 28.08% for KG construction and a precision of 57.89%, a recall of 48.61%, and an F1-score of 52.8% for entity recognition. The introduction of CoT in zero-shot learning yielded a slightly lower performance and improved only the ROUGE-2 score to 30.54%.

In the few-shot learning approaches, the application of a fixed set of examples resulted in improved precision, recall, F1-score, and ROUGE-2 on the graph level as well as precision, recall and F1-score on the entity level, indicating that providing the model with specific examples enhances performance. The random selection strategy showed similar results to the fixed selection, while category-based selection significantly outperformed other methods, achieving a precision of 37.93%, recall of 33.55%, and the highest F1-score and ROUGE-2 score among the few-shot learning, with 35.61% and 49.35% respectively. The similarity-based selection method did not perform as well as the category-based but did surpass the fixed and random strategies in terms of the ROUGE-2 score. The limitation may stem from the similarity function employed, which might not capture adequate semantic information. Different embedding methods and similarity functions could be explored. Notably, our proposed approach, few-shot learning with implicit and explicit verbal self-reflection, denoted as SR_i and SR_e respectively, further improved the results. The implicit method (SR_i) achieved the highest precision of 41.01% and a strong F1-score of 37.10%, while the explicit method (SR_i) resulted in the highest recall of 34.88% and the highest ROUGE-2 score of 49.55%. However, in terms of entity recognition, self-reflection approaches yielded a slightly lower performance.

In contrast, the pipeline-based approach with zero-shot learning, which did not utilize the few-shot learning advantages, resulted in the lowest scores across all metrics, with a precision of 23.29%, recall of 16.39%, F1-score of 19.24%, and ROUGE-2 score of 28.68% for KG construction.

4.3 Discussions

Applying the few-shot learning of GPT-4 significantly improves the performance of KGC compared to the zero-shot baseline. Specifically, the category-based example selection strategy outperforms others, achieving the highest precision, recall, F1-score, and ROUGE-2 score in most cases, which indicates that the relevance and representativeness of examples play a crucial role in enhancing the model performance. Our proposed self-reflection mechanism, both implicit and explicit, further enhances the model's capability to generate accurate and reliable KG triples. The implicit self-reflection method achieved the highest precision, while the explicit self-reflection method resulted in the highest recall and ROUGE-2 score. This shows the potential of self-reflection in improving the in-context learning process by enabling the model to learn from its mistakes. The zero-shot learning approach, as well as with Chain-of-Thought (CoT) prompting, falls short in performance compared to few-shot learning methods, which shows the importance of providing contextual examples to guide the model outputs effectively.

The similarity-based selection strategies in our evaluation did not achieve the expected performance, the underperformance of similarity-based selection could be the limitations of the similarity measurement technique used. In our implementation, we utilized SentenceBERT with the paraphrase-MiniLM-L12-v2 model to calculate cosine similarities. While this model is effective for general sentence embeddings, it may not capture the semantic relationships required for effective KGC, particularly in specialized domains. Another reason is that input text from different categories could be similar, but their relation lists are different, such as Building, Geographic Location, and Transport. Few-shot learning is still dependent on the quality and diversity of the labeled data pool. In real-world scenarios, the availability of such labeled data can be a limitation. One of the challenges in implementing our proposed methods is the high cost associated with calling the OpenAI API, which limits the scalability of our approach. Moreover, our evaluation primarily focuses on precision, recall, F1-score, and ROUGE-2 scores. While these metrics provide a good indication of performance, they may not fully capture the semantic accuracy and usefulness of the constructed knowledge graphs. Future work should explore more sophisticated and holistic evaluation metrics. Besides, it's hard to accurately extract desired entities, i.e., entities ranging from a single character to a phrase may be extracted, resulting in low precision and error propagation.

5 CONCLUSION

In this paper, we explore methods that leverage the in-context learning capabilities of LLMs, especially GPT-4, for instruction-driven adaptive KG construction and propose a self-reflection mechanism that reflects on errors from examples. Experiments show that example selection is essential for few-shot learning of LLMs and demonstrate the superiority of our proposed method in terms of KG construction. We hope the methodology, evaluation, and findings presented in this paper will serve as a good starting point to explore effective interactions between LLMs and KGs. Our study may also motivate data management researchers and experts interested in exploring the synergy between LLMs and KGs to discuss this topic further.

In future work, we would like to investigate more sophisticated metrics and algorithms for selecting examples and further refine the quality of examples used for in-context learning, including exploration beyond BERTScore and OpenAI embedding to newer semantic similarity measures. Furthermore, as the labelled data pool is not always available, we would like to explore the possibility of using semi-supervised methods to build the annotated data pool.

ACKNOWLEDGMENTS

This work was supported by the German Ministry for Research and Education (BMBF) project WestAI (Grant no. 01IS22094D).

REFERENCES

[1] 2023. Chain-of-Thought Prompt Distillation for Multimodal Named Entity and Multimodal Relation Extraction. *CoRR* abs/2306.14122 (2023). https://doi.org/10.48550/ARXIV.2306.14122 arXiv:2306.14122 Withdrawn.

[2] Dhananjay Ashok and Zachary C. Lipton. 2023. PromptNER: Prompting For Named Entity Recognition. *CoRR* abs/2305.15444 (2023). https://doi.org/10.48550/ARXIV.2305.15444 arXiv:2305.15444

[3] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary G. Ives. 2007. DBpedia: A Nucleus for a Web of Open Data. In *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007 (Lecture Notes in Computer Science)*, Karl Aberer, Key-Sun Choi, Natasha Fridman Noy, Dean Allemang, Kyung-Il Lee, Lyndon J. B. Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber, and Philippe Cudré-Mauroux (Eds.), Vol. 4825. Springer, 722–735. https://doi.org/10.1007/978-3-540-76298-0_52

[4] Junwei Bao, Nan Duan, Zhao Yan, Ming Zhou, and Tiejun Zhao. 2016. Constraint-Based Question Answering with Knowledge Graph. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, Nicoletta Calzolari, Yuji Matsumoto, and Rashmi Prasad (Eds.). ACL, 2503–2514. https://aclanthology.org/C16-1236/

[5] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html

[6] Yilong Chen, Shiyao Cui, Kun Huang, Shicheng Wang, Chuanyu Tang, Tingwen Liu, and Binxing Fang. 2023. Improving Adaptive Knowledge Graph Construction via Large Language Models with Multiple Views. In *China Conference on Knowledge Graph and Semantic Computing*. Springer, 273–284.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, 4171–4186. https://doi.org/10.18653/V1/N19-1423

[8] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. 2023. A Survey for In-context Learning. *CoRR* abs/2301.00234 (2023). https://doi.org/10.48550/ARXIV.2301.00234

arXiv:2301.00234

[9] Jenny Rose Finkel, Trond Grenager, and Christopher D. Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *ACL 2005, 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 25-30 June 2005, University of Michigan, USA*, Kevin Knight, Hwee Tou Ng, and Kemal Oflazer (Eds.). The Association for Computer Linguistics, 363–370. https://doi.org/10.3115/1219840.1219885

[10] Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making Pre-trained Language Models Better Few-shot Learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.). Association for Computational Linguistics, 3816–3830. https://doi.org/10.18653/V1/2021.ACL-LONG.295

[11] Bernal Jimenez Gutierrez, Nikolas McNeal, Clayton Washington, You Chen, Lang Li, Huan Sun, and Yu Su. 2022. Thinking about GPT-3 In-Context Learning for Biomedical IE? Think Again. In *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (Eds.). Association for Computational Linguistics, 4497–4512. https://doi.org/10.18653/V1/2022.FINDINGS-EMNLP.329

[12] Qing Huang, Yanbang Sun, Zhenchang Xing, Min Yu, Xiwei Xu, and Qinghua Lu. 2024. API Entity and Relation Joint Extraction from Text via Dynamic Prompt-tuned Language Model. *ACM Trans. Softw. Eng. Methodol.* 33, 1 (2024), 6:1–6:25. https://doi.org/10.1145/3607188

[13] Bin Ji. 2023. VicunaNER: Zero/Few-shot Named Entity Recognition using Vicuna. *CoRR* abs/2305.03253 (2023). https://doi.org/10.48550/ARXIV.2305.03253 arXiv:2305.03253

[14] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. 2022. A Survey on Knowledge Graphs: Representation, Acquisition, and Applications. *IEEE Trans. Neural Networks Learn. Syst.* 33, 2 (2022), 494–514. https://doi.org/10.1109/TNNLS.2021.3070843

[15] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. https://openreview.net/forum?id=H1eA7AEtvS

[16] Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*. Association for Computational Linguistics, Barcelona, Spain, 74–81. https://aclanthology.org/W04-1013

[17] Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona T. Diab, Veselin Stoyanov, and Xian Li. 2022. Few-shot Learning with Multilingual Generative Language Models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (Eds.). Association for Computational Linguistics, 9019–9052. https://doi.org/10.18653/V1/2022.EMNLP-MAIN.616

[18] Jiaying Liu, Jing Ren, Wenqing Zheng, Lianhua Chi, Ivan Lee, and Feng Xia. 2020. Web of Scholars: A Scholar Knowledge Graph. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, Jimmy X. Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu (Eds.). ACM, 2153–2156. https://doi.org/10.1145/3397271.3401405

[19] Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What Makes Good In-Context Examples for GPT-3?. In *Proceedings of Deep Learning Inside Out: The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures, DeeLIO@ACL 2022, Dublin, Ireland and Online, May 27, 2022*, Eneko Agirre, Marianna Apidianaki, and Ivan Vulic (Eds.). Association for Computational Linguistics, 100–114. https://doi.org/10.18653/V1/2022.DEELIO-1.10

[20] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the Middle: How Language Models Use Long Contexts. *CoRR* abs/2307.03172 (2023). https://doi.org/10.48550/ARXIV.2307.03172 arXiv:2307.03172

[21] Harsha Nori, Yin Tat Lee, Sheng Zhang, Dean Carignan, Richard Edgar, Nicolò Fusi, Nicholas King, Jonathan Larson, Yuanzhi Li, Weishung Liu, Renqian Luo, Scott Mayer McKinney, Robert Osazuwa Ness, Hoifung Poon, Tao Qin, Naoto Usuyama, Chris White, and Eric Horvitz. 2023. Can Generalist Foundation Models Outcompete Special-Purpose Tuning? Case Study in Medicine. *CoRR* abs/2311.16452 (2023). https://doi.org/10.48550/ARXIV.2311.16452 arXiv:2311.16452

[22] Ciyuan Peng, Feng Xia, Mehdi Naseriparsa, and Francesco Osborne. 2023. Knowledge Graphs: Opportunities and Challenges. *Artif. Intell. Rev.* 56, 11 (2023), 13071–13102. https://doi.org/10.1007/S10462-023-10465-9

[23] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training. (2018).

[24] 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. *J. Mach. Learn. Res.* 21 (2020), 140:1–140:67. http://jmlr.org/papers/v21/20-074.html

[25] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. 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 2019, Hong Kong, China, November 3-7, 2019*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, 3980–3990. https://doi.org/10.18653/V1/D19-1410

[26] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR* abs/1910.01108 (2019). arXiv:1910.01108 http://arxiv.org/abs/1910.01108

[27] Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilic, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, and et al. 2022. BLOOM: A 176B-Parameter Open-Access Multilingual Language Model. *CoRR* abs/2211.05100 (2022). https://doi.org/10.48550/ARXIV.2211.05100 arXiv:2211.05100

[28] Stacy Schiff. 2006. Know it all. *The New Yorker* 31, 07 (2006).

[29] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (Eds.). http://papers.nips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html

[30] Wee Meng Soon, Hwee Tou Ng, and Chung Yong Lim. 2001. A Machine Learning Approach to Coreference Resolution of Noun Phrases. *Comput. Linguistics* 27, 4 (2001), 521–544. https://doi.org/10.1162/089120101753342653

[31] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007*, Carey L. Williamson, Mary Ellen Zurko, Peter F. Patel-Schneider, and Prashant J. Shenoy (Eds.). ACM, 697–706. https://doi.org/10.1145/1242572.1242667

[32] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. *CoRR* abs/2302.13971 (2023). https://doi.org/10.48550/ARXIV.2302.13971 arXiv:2302.13971

[33] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem 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, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. *CoRR* abs/2307.09288 (2023). https://doi.org/10.48550/ARXIV.2307.09288 arXiv:2307.09288

[34] Milena Trajanoska, Riste Stojanov, and Dimitar Trajanov. 2023. Enhancing Knowledge Graph Construction Using Large Language Models. *CoRR* abs/2305.04676 (2023). https://doi.org/10.48550/ARXIV.2305.04676 arXiv:2305.04676

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). 5998–6008. https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[36] Somin Wadhwa, Silvio Amir, and Byron C. Wallace. 2023. Revisiting Relation Extraction in the era of Large Language Models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, 15566–15589. https://doi.org/10.18653/V1/2023.ACL-LONG.868

[37] Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. 2023. GPT-NER: Named Entity Recognition via Large Language Models. *CoRR* abs/2304.10428 (2023). https://doi.org/10.48550/ARXIV.2304.10428 arXiv:2304.10428

[38] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. KGAT: Knowledge Graph Attention Network for Recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, Ankur Teredesai, Vipin Kumar, Ying Li, Rómer Rosales, Evimaria Terzi, and George Karypis (Eds.). ACM, 950–958. https://doi.org/10.1145/3292500.3330989

[39] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net. https://openreview.net/forum?id=1PL1NIMMrw

[40] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (Eds.). http://papers.nips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html

[41] 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. *CoRR* abs/2302.10205 (2023). https://doi.org/10.48550/ARXIV.2302.10205 arXiv:2302.10205

[42] Benfeng Xu, Quan Wang, Yajuan Lyu, Dai Dai, Yongdong Zhang, and Zhendong Mao. 2023. 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), ACL 2023, Toronto, Canada, July 9-14, 2023*, Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, 8186–8207. https://doi.org/10.18653/V1/2023.ACL-LONG.455

[43] Xin Xu, Yuqi Zhu, Xiaohan Wang, and Ningyu Zhang. 2023. How to Unleash the Power of Large Language Models for Few-shot Relation Extraction?. In *Proceedings of The Fourth Workshop on Simple and Efficient Natural Language Processing, SustaiNLP 2023, Toronto, Canada (Hybrid), July 13, 2023*, Nafise Sadat Moosavi, Iryna Gurevych, Yufang Hou, Gyuwan Kim, Young Jin Kim, Tal Schuster, and Ameeta Agrawal (Eds.). Association for Computational Linguistics, 190–200. https://doi.org/10.18653/V1/2023.SUSTAINLP-1.13

[44] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net. https://openreview.net/forum?id=WE_vluYUL-X

[45] Alexander Yates, Michele Banko, Matthew Broadhead, Michael J. Cafarella, Oren Etzioni, and Stephen Soderland. 2007. TextRunner: Open Information Extraction on the Web. In *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, April 22-27, 2007, Rochester, New York, USA*, Candace L. Sidner, Tanja Schultz, Matthew Stone, and ChengXiang Zhai (Eds.). The Association for Computational Linguistics, 25–26. https://aclanthology.org/N07-4013/

[46] Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel Methods for Relation Extraction. *J. Mach. Learn. Res.* 3 (2003), 1083–1106. http://jmlr.org/papers/v3/zelenko03a.html

[47] Lingfeng Zhong, Jia Wu, Qian Li, Hao Peng, and Xindong Wu. 2024. A Comprehensive Survey on Automatic Knowledge Graph Construction. *ACM Comput. Surv.* 56, 4 (2024), 94:1–94:62. https://doi.org/10.1145/3618295

[48] Guodong Zhou and Jian Su. 2002. Named Entity Recognition using an HMM-based Chunk Tagger. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*. ACL, 473–480. https://doi.org/10.3115/1073083.1073163