Semantic Web -1 (2024) 1–34
DOI 10.3233/SW-243719
IOS Press
CORRECTED PROOF

Testing prompt engineering methods for knowledge extraction from text

Fina Polat [a,*], Ilaria Tiddi [b] and Paul Groth [a]

[a] *Intelligent Data Engineering Lab (INDElab), Informatics Institute, University of Amsterdam, The Netherlands*
E-mails: f.yilmazpolat@uva.nl, p.t.groth@uva.nl

[b] *Artificial Intelligence, Faculty of Science, Vrije Universiteit Amsterdam, The Netherlands*
E-mail: i.tiddi@vu.nl

Editors: Sanju Tiwari, UAT, Mexico; Nandana Mihindukulasooriya, MIT-IBM Watson AI Lab, USA; Francesco Osborne, KMi, The Open University, United Kingdom; Dimitris Kontokostas, Medidata, Greece; Jennifer D'Souza, TIB, Germany; Mayank Kejriwal, University of Southern California, USA
Solicited reviews: Dimitris Kontokostas, Medidata Knowledge Graph, Greece; four anonymous reviewers

Abstract. The capabilities of Large Language Models (LLMs,) such as Mistral 7B, Llama 3, GPT-4, present a significant opportunity for knowledge extraction (KE) from text. However, LLMs' context-sensitivity can hinder obtaining precise and task-aligned outcomes, thereby requiring prompt engineering. This study explores the efficacy of five prompt methods with different task demonstration strategies across 17 different prompt templates, utilizing a relation extraction dataset (RED-FM) with the aforementioned LLMs. To facilitate evaluation, we introduce a novel framework grounded in Wikidata's ontology. The findings demonstrate that LLMs are capable of extracting a diverse array of facts from text. Notably, incorporating a simple instruction accompanied by a task demonstration – comprising three examples selected via a retrieval mechanism – significantly enhances performance across Mistral 7B, Llama 3, and GPT-4. The effectiveness of reasoning-oriented prompting methods such as Chain-of-Thought, Reasoning and Acting, while improved with task demonstrations, does not surpass alternative methods. This suggests that framing extraction as a reasoning task may not be necessary for KE. Notably, task demonstrations leveraging examples selected via retrieval mechanisms facilitate effective knowledge extraction across all tested prompting strategies and LLMs.

Keywords: Prompt engineering, Generative Knowledge Extraction, Ontology based evaluation, GPT-4, Mistral 7B, Llama-3, Wikidata

1. Introduction

Knowledge Extraction (KE), or Knowledge Triple Extraction, aims to identify entities and their semantic relations. KE is a crucial task towards automatically constructing large-scale knowledge graphs [20].

Large Language Models (LLMs) have shown state-of-the-art performance on knowledge extraction tasks [12,15, 32]. Current leading models use a generative approach where sequence-to-sequence models are trained end-to-end to go from raw texts to <subject–predicate–object> triples. While generative knowledge extraction approaches rely on fine-tuning models, recent work [17] has suggested that there is potential in using in-context learning (ICL) [3] to perform knowledge extraction. ICL leverages the concept of "learning by demonstration", a method where the

*Corresponding author. E-mail: f.yilmazpolat@uva.nl.

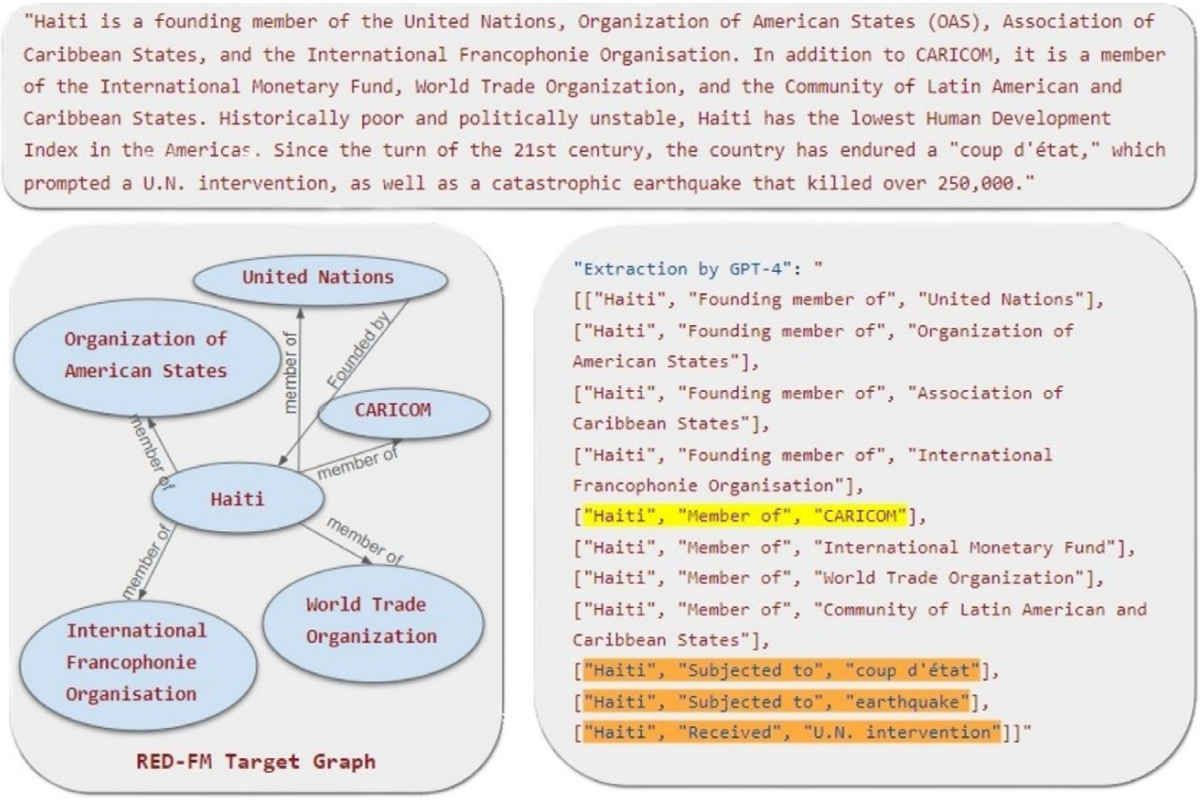

Fig. 1. A knowledge extraction example: RED-FM target graph vs. triples extracted by GPT-4.

model is trained to recognize and replicate tasks by examining provided examples. This phenomenon allows the model to mimic task-specific behavior by adjusting its generated responses to match the demonstrated examples, regardless of the task or LLM involved. This can be beneficial when having large amount of parameters (as LLMs do), as it eliminates the need to train or fine-tune models. By providing a well devised input sequence (e.g. a textual prompt), LLMs can therefore learn to perform knowledge extraction tasks. However, the question is how to devise such a good input prompt, and this is where prompt engineering comes into play.

To answer this question, our study investigates which state-of-the-art prompt engineering methods work best for the task of knowledge extraction. We adapt chain-of-thought [31], reasoning and acting [33], self-consistency [30], and generated knowledge [18] with different task demonstration strategies to the context of knowledge extraction and test their respective performance with three potent generative large language models, i.e. GPT-4 [22], Mistral 7B [11], Llama 3 [1]. To evaluate these methods, we utilize a relation extraction dataset, i.e. RED-FM [10], in an Open Information Extraction setup. Specifically, given a piece of text, we aim to extract all potential triples (e.g. relations) from the text, without supplying any predetermined labels nor imposing constraints within the prompt as in Closed Information Extraction settings [17].

Figure 1 displays an example of our task performed by the GPT-4 [22] model using a simple instruction followed by a task demonstration with three examples selected by a retrieval mechanism. The input text and the target triples are both taken from RED-FM. According to the classical, strict match-based evaluation metrics measuring the exact correspondence between the extracted triples and a predefined set of target triples, there is only one correct extraction – the one we highlighted in yellow in the figure. However, the LLM is capable of extracting finer-grained relations from the given text as, for instance, the *founding member of* relation instead of *member of*. Furthermore, it is capable of extracting event facts, highlighted in orange in the figure, which are out of the scope of the original RED-FM task. These strict evaluation metrics and language variability are a bottleneck for the evaluation of the

LLM's extractions, as they complicate the task of devising evaluation criteria that are both robust and universally applicable across different contexts and applications [7].

To overcome these limitations, our study presents a novel evaluation approach rooted in Wikidata [28] and the semantics of the Wikidata schema or ontology. This method involves a comprehensive examination of all extracted triples against Wikidata. In order to identify the most effective prompt engineering method for extracting knowledge, the validity of triples extracted via the prompt engineering methods is assessed by cross-referencing them with the knowledge retrieved from Wikidata. This benchmarking exercise provides a quantifiable metric to ascertain the accuracy and reliability of the factual knowledge extracted by LLMs.

In summary, this work provides two main contributions:

1. The adaptation and evaluation of state-of-the-art prompt engineering methods with different task demonstration strategies in the context of KE using three LLMs (Mistral 7B [11], Llama 3 [1], GPT-4 [22]).
2. The introduction of an evaluation protocol based on Wikidata to asses the performance of open information extraction approaches.

The complete data utilized in this study, as well as the source code which underpins the paper, can be accessed from the GitHub repository .[1]

2. Background

2.1. Generative knowledge extraction

Extracting knowledge triples from text has been a continuous research topic for the Natural Language Processing (NLP) and Semantic Web (SW) communities [20]. Traditionally this task has been approached as a two-step problem. First, the entities are extracted from text as in Named Entity Recognition (NER). Second, Relation Classification (RC) checks whether there exists any pairwise relation between the extracted entities [35,37]. This two-step approach requires additional annotations to identify which entities share a relation.

Recent approaches tackle both tasks simultaneously as a sequence-to-sequence learning problem referred to as End-to-End Relation Extraction (RE). In this approach, a model is trained simultaneously on both NER and RC objectives [9,12,13]. Training both tasks simultaneously in multi-task setups results in improving performance on the end-to-end RE task without additional annotation. However, it still requires significant amount of training data.

Scaling up language models has improved task-agnostic few-shot performance [3]. ICL is an emergent capability of Large Language Models (LLMs) in particular those with billions of parameters such as GPT-3 [3], LLaMDA [26], PaLM [5], LLaMA [27], and GPT-4 [22]. The capability became prominent in GPT models particularly starting from GPT-3 [3]. ICL is a way to use language models to learn tasks given only a few examples without any additional training or finetuning. The model performs a task just by conditioning on prompts, without optimizing any parameters. Our approach focuses on ICL (i.e. prompting, few-shot learning), which combines the capabilities of LLMs with the contextual information available in the text.

2.2. Prompt engineering

Prompts serve as instructive cues for LLMs, directing them to generate desired outputs. They can range from straightforward direct questions, such as "What are the capital cities in the European Union?" to more instructional formulations like "List the capital cities in the European Union." It is crucial to recognize that LLMs, being inherently context-sensitive, may produce different responses based on the specific formulation of the prompt. Prompt engineering focuses on the development and optimization of prompts [19]. This practice aims to enhance the efficiency of LLMs across a diverse spectrum of tasks and applications. By carefully crafting and refining prompts, researchers and practitioners seek to harness the full potential of LLMs, tailoring their responses to align with the intricacies of various tasks and objectives.

[1] https://github.com/FinaPolat/Prompt-Engineering-for-KE

A comprehensive prompt comprises multiple elements, including a task description, context, a request or question, and examples. Certain components of a prompt, such as the task description, can be expressed in various ways, introducing a layer of flexibility. On the other hand, the selection of context and/or examples is inherently dynamic, involving a retrieval mechanism for optimal relevance. In order to incorporate different components in a coherent way, designing a prompt template is essential [19]. For a more in-depth exploration of prompting strategies and associated architectures in NLP, the reader is directed to the survey conducted by Min et al. [21].

2.3. Evaluation challenges in generative approaches

The open-ended nature of open information extraction using generative approaches creates a challenge for automatic evaluation [7]. The expressiveness of LLMs as illustrated in Fig. 1 coupled with the open-endedness of extraction of knowledge triples makes evaluation non-trivial. The reliance on exact matches to predefined targets becomes impractical due to the variability in expression present in generated text. Prior research has predominantly employed a strict evaluation methodology, necessitating exact matches between generated triples and references. This approach proves suitable for evaluating smaller conditional generation models designed for RE such as REBEL [9] that is based on BART [15]. These models, having undergone extensive finetuning on large datasets, and hence consistently produce standardized outputs.

In contrast, larger and more expressive language models like Llama 3 exhibit the capacity to generate a diverse array of output formats that convey similar content to targets as demonstrated in Fig. 1. Unlike traditional approaches where NER and RE models classify or label input tokens, generative language models such as Mistral 7B generate new tokens from a large vocabulary, rather than selecting from a pre-defined set of classes [29]. Given the open-ended nature of outputs from such models and the challenge of aligning them with predefined standards, practitioners often opt for human evaluation. Although this approach is time-consuming and more expensive, it provides a qualitative assessment that better displays the performance of the generative model and the quality and correctness of the generated triples [17,29].

3. Prompt engineering methods

Our approach involves adapting prominent prompt engineering methods to KE with the goal of identifying the most effective prompting strategy for it. Prompt engineering methods generally follow a similar pattern. Initially, a prompt template is formulated, defining the structure and placeholders. Subsequently, depending on the specific engineering method, input data and examples are inserted into the template. Below we describe the prompt engineering methods used in this study.

3.1. Simple instruction: Zero-shot, one-shot, few-shots prompts

LLMs acquire knowledge from text corpora during training, as outlined in [23]. The pre-training objective for LLMs typically involves minimizing contextual word prediction errors on extensive corpora. However, achieving an understanding of user intent and effectively following instructions represents a more intricate Natural Language Understanding (NLU) task. This level of understanding does not automatically emerge as a result of scaling; rather, it necessitates a deliberate alignment with user intent, as highlighted in [36].

Prompting with a simple instruction without any task demonstration leverages a LLM's acquired NLU capabilities and, to some extent, its internal knowledge to execute a specified task. We call this type of instruction only prompts "zero-shot" prompts. Typically, a zero-shot prompt comprises a direct instruction. However, ICL performs well when a prompt is combined with a task description and an example showing the execution of the described task. ICL allows for an expansion in the number of examples within the constraints of the context window. This approach enhances the model's ability to execute tasks within a given context. In our zero-shot test setting, there is no task description or any task demonstration incorporated into the prompt; instead it is only a simple direct instruction followed by the input text and a signal phrase, i.e. Your answer. The signal phrase is included to demarcate the conclusion of the prompt. The following example illustrates a zero-shot prompt.

```
Extract knowledge triples from the text. Return the triples in JSON format.
Text: The Porsche Panamera is a mid/full-sized luxury vehicle (E-segment in Europe)
manufactured by the German automobile manufacturer Porsche. It is front-engined and
has a rear-wheel-drive layout, with all-wheel drive versions also available.
Your answer:
```

Then, we add task demonstration into the prompts, and name versions according to the number of examples used. One-shot prompts have one canonical example along with the instruction. The example, ie. "Text: The Amazon River flows through Brazil and Peru. {"Triples": [["Amazon River", "country", "Brazil"], ["Amazon River", "country", "Peru"]]}" remains the same for all of the inputs. The following example illustrates a one-shot prompt.

```
Extract knowledge triples from the text. Return the triples in JSON format.
Here is an example.
Text: The Amazon River flows through Brazil and Peru.
Your answer: {"Triples": [["Amazon River", "country", "Brazil"], ["Amazon River",
"country", "Peru"]]}
Text: The Porsche Panamera is a mid/full-sized luxury vehicle (E-segment in Europe)
manufactured by the German automobile manufacturer Porsche. It is front-engined and
has a rear-wheel-drive layout, with all-wheel drive versions also available.
Your answer:
```

One-shot prompting with a fixed example provides a reference point to understand the intended task and output format, and execute it accordingly. Few-shot prompts retain the same instruction and the first example, but the number of examples is extended to three (canonical examples) for the few-shot test. Much like the one-shot setting, these examples remain uniform across all data points. The following example displays a few-shots prompt.

```
Extract knowledge triples from the text. Return the triples in JSON format.
Here are a few examples.
Text: The Amazon River flows through Brazil and Peru.
Your answer: {"Triples": [["Amazon River", "country", "Brazil"], ["Amazon River",
"country", "Peru"]]}
Text: COVID-19 symptoms include fever, cough, and shortness of breath.
Your Answer: {"Triples": [["COVID-19", "symptom", "fever"], ["COVID-19", "symptom",
"cough"], ["COVID-19", "symptom", "shortness of breath"]]}
Text: The American Civil War took place from 1861 to 1865.
Your answer: {"Triples": [["American Civil War", "start date", "1861"],
["American Civil War", "end date", "1865"]]}
Text: The Porsche Panamera is a mid/full-sized luxury vehicle (E-segment in Europe)
manufactured by the German automobile manufacturer Porsche. It is front-engined and
has a rear-wheel-drive layout, with all-wheel drive versions also available.
Your answer:
```

3.2. Selecting examples for task demonstration

LLMs shows significant capability in learning from the context within prompts, underscoring the importance of incorporating relevant examples. One-shot and few-shot prompts exploit canonical examples that remains the same for all of the input instances regardless of their relevance. However, using relevant examples contributes ICL capabilities of LLMs. One notable advantage of ICL is its diminished dependence on extensive amounts of annotated data. In scenarios such as our test setting, training data remains untapped. With a retrieval mechanism, however, training data can be a valuable source for finding contextually relevant examples to input instances, as a result of improving the performance of ICL.

The retrieval of examples from the training set for a generation task can be facilitated through several search mechanisms. The choice of a specific example retriever is non-trivial, as it significantly impacts the retrieved examples and their subsequent contribution to task execution. For our experiments, we employ a state-of-the-art example retrieval mechanism based on maximal marginal relevance [34]. The Maximal Marginal Relevance (MMR) [4] approach selects exemplars that are both relevant as well as diverse. The underlying rationale for the selection of MMR is reducing redundancy in the selected examples. Additionally, a diverse set of exemplars is more likely to showcase complementary signal that is required to extract accurate triples from the input text.

All examples in the following prompt engineering methods are selected by the MMR example retriever [34].

3.2.1. Incorporating retrieved examples into the prompts

The ability of LLMs accessing to knowledge and precisely manipulating it remains limited. To address this limitation, a differentiable access mechanism to explicit non-parametric memory can be employed. Retrieval Augmented Generation (RAG) is a general-purpose fine-tuning recipe for LLMs which combines pre-trained parametric and non-parametric memories for language generation [16]. The RAG formulation involves the convergence of key components, namely a knowledge base/documents, an embedder, a retriever, and a generator. In this process, the knowledge base undergoes an embedding operation facilitated by the embedder. Then, the retriever conducts search on the embedded knowledge base and retrieves the context similar to the input query. Finally, the retrieved context combined with the input is fed into the generation module.

Our approach does not apply RAG formulation directly but instead it augments prompt content by retrieving an example/examples from a randomly selected subset of training data. For that, we randomly sample 300 instances from training data. Test inputs and training sample for example retrieval are embedded using an open source state-of-the-art instruction-finetuned text embedding model [25]. We distinguish this type of prompts which contains task demonstrations selected by a retrieval mechanism from those with canonical examples, and call this type of prompts RAG prompts. For example, an example retrieved by the MMR [34] gets incorporated into the prompt as the task demonstration for one-shot RAG prompts, and three examples for few-shot RAG prompts. The following example shows a one-shot RAG prompt.

```
Extract knowledge triples from the text.
Here is an example:
Text: Audi AG () is a German automobile manufacturer that designs, engineers,
produces, markets and distributes luxury vehicles. Audi is a subsidiary of
Volkswagen Group and has its roots at Ingolstadt, Bavaria, Germany. Audi
vehicles are produced in nine production facilities worldwide.
Your answer: {"Triples": [["Audi", "Country", "German"], ["Audi", "Owned by",
"Volkswagen Group"], ["Audi", "Headquarters location", "Ingolstadt"]]}
The example ends here.
Text: The Porsche Panamera is a mid/full-sized luxury vehicle (E-segment in Europe)
manufactured by the German automobile manufacturer Porsche. It is front-engined and
has a rear-wheel-drive layout, with all-wheel drive versions also available.
Your answer:
```

The difference between one-shot/few-shot and one-shot RAG/few-shot RAG prompts lays in the incorporation of examples. The retrieving module dynamically selects the most relevant examples from the sample. The following example displays the effectiveness of the retrieval module on a few-shots RAG prompt.

```
Extract knowledge triples from the text. Here are a few examples:
Text: Audi AG () is a German automobile manufacturer that designs, engineers,
produces, markets and distributes luxury vehicles. Audi is a subsidiary of
Volkswagen Group and has its roots at Ingolstadt, Bavaria, Germany. Audi vehicles
are produced in nine production facilities worldwide.
Your answer: {"Triples": [["Audi", "Country", "German"], ["Audi", "Owned by",
"Volkswagen Group"], ["Audi", "Headquarters location", "Ingolstadt"]]}
Text: Atlas V is an expendable launch system and the fifth major version in the
```

```
Atlas rocket family. It was originally designed by Lockheed Martin, now being
operated by United Launch Alliance (ULA), a joint venture between Lockheed Martin
and Boeing. Atlas V is also a major NASA launch vehicle.
Your answer: {"Triples": [["Atlas V", "Manufacturer", "United Launch Alliance"]]}
Text: The main product lines from Altera were the Stratix, mid-range Arria, and
lower-cost Cyclone series system on a chip FPGAs, the MAX series complex program-
mable logic device and non-volatile FPGAs, Intel Quartus Prime design software,
and Enpirion PowerSoC DC-DC power solutions.
Your answer: {"Triples": [["FPGAs", "Manufacturer", "Altera"]]}
Examples end here.
Text: The Porsche Panamera is a mid/full-sized luxury vehicle (E-segment in Europe)
manufactured by the German automobile manufacturer Porsche. It is front-engined
and has a rear-wheel-drive layout, with all-wheel drive versions also available.
Your answer:
```

3.3. Chain-of-thought prompts

A chain of thought is a series of intermediate natural language reasoning steps that lead to the final output, this approach is referred as chain-of-thought prompting (CoT). CoT prompting aims to enable LLMs' complex reasoning capabilities through intermediate reasoning steps. Empirical evidence suggests that this technique facilitates improved performance across diverse domains, including arithmetic problem-solving, commonsense reasoning, and symbolic logic tasks. The efficacy of CoT prompting in enhancing the reasoning capabilities of LLMs is studied in recent research [31].

The exploration of CoT prompting within the context of KE represents a novel strategy aimed at enhancing the ability of LLMs in extracting structured information from text. The premise underpinning this methodology is that by guiding LLMs to reason explicitly about entities, their respective types, and the relations between them, the quality of the extracted knowledge triples may be improved.

In a zero-shot CoT setting, along with the direct instruction the task description of KE is also presented: "A knowledge triple consists of three elements: subject – predicate – object. Subjects and objects are entities and the predicate is the relation between them." This can also be considered as a concept definition which is different from a direct instruction: "Extract knowledge triples from the text". Then, the LLM is instructed to "think step by step," thereby engaging in a sequential reasoning process without prior exposure to explicit examples. In one-shot and few-shot settings, we only employ retrieved examples to demonstrate the execution of the task in three intermediate steps since empirical evidence suggest that retrieved examples contributes to model performance more than fixed canonical examples. These three steps are widely used in traditional KE pipelines, i.e., entity extraction, entity typing, and relation extraction. RED-FM contains entity type annotation along with entities and relations. We use a template to convert entity type annotation into natural language descriptions. An example of our CoT application on KE is demonstrated below.

```
Your task is extracting knowledge triples from text. A knowledge triple consists
of three elements: subject - predicate - object. Subjects and objects are entities
and the predicate is the relation between them. Before extracting triples, let's
think step by step.
Here is an example:
Text: Audi AG () is a German automobile manufacturer that designs, engineers,
produces, markets and distributes luxury vehicles. Audi is a subsidiary of
Volkswagen Group and has its roots at Ingolstadt, Bavaria, Germany. Audi vehicles
are produced in nine production facilities worldwide.
Let's extract the entities first. Here is the list of the entities in this text:
["Audi", "German", "automobile manufacturer", "luxury vehicle", "Volkswagen Group",
"Ingolstadt", "Bavaria"]
```

```
What do you know about the entities?
[“Automobile manufacturer is a/an concept.”, “Luxury vehicle is a/an concept.”]
Now we think about the potential relations between these entities:
[“country”, “owned by”, “headquarters location”]
Now we can extract the triples:
[[“Audi”, “Country”, “German”], [“Audi”, “Owned by”, “Volkswagen Group”],
[“Audi”, “Headquarters location”, “Ingolstadt”]]
Example ends here.
Extract the triples from the following text thinking step by step.
Text: The Porsche Panamera is a mid/full-sized luxury vehicle (E-segment in Europe)
manufactured by the German automobile manufacturer Porsche. It is front-engined and
has a rear-wheel-drive layout, with all-wheel drive versions also available.
Your answer:
```

3.4. Self-consistency prompts

CoT's initial implementations rely on naive greedy decoding, which does not account for exploring alternative reasoning pathways or correcting misconceptions during the reasoning process. To improve upon this, the concept of self-consistency prompting has been introduced [30]. Self-consistency prompting allows LLMs to consider multiple reasoning paths and offers a mechanism for identifying and correcting errors. This method aims to enhance the decision-making process of LLMs by shifting from the original CoT's simplistic decoding strategy to a more comprehensive reasoning approach. The advantages and implications of self-consistency prompting are further explored in [30].

In the application of self-consistency prompting to KE, we enhance prompts by incorporating an additional layer of reflection into the task execution sequence. In zero-shot setting, the following instruction is used: “First, think about entities and relations that you want to extract from the text. Then, look at the potential triples. Think like a domain expert and check the validity of the triples. Filter out the invalid triples. Return the valid triples in JSON format.” In one-shot and few-shot settings, execution of the task is demonstrated on the retrieved examples.

The main distinction between CoT and the self-consistency prompts lies in the request for the creation of a preliminary list for potential extractions and then critically evaluating this list. In one-shot/few-shot self-consistency prompts, an erroneous triple in the drafts is deliberately included. Then, the incorrect triple is shown to the LLM, accompanied by an explanation. The following example shows the full application of self-consistency prompting on KE.

```
Your task is extracting knowledge triples from text.
A knowledge triple consists of three elements: subject - predicate - object.
Subjects and objects are entities and the predicate is the relation between them.
Let’s use an example:
Text: The Airbus A380 is a wide-body aircraft manufactured by Airbus. It is the
world’s largest passenger airliner. Let’s extract the entities first.
Here is the list of the entities in this text:
[“Airbus A380”, “380”, “wide-body aircraft”, “Airbus”]
What do you know about the entities?
[“380 is a/an number.”, “Wide-body aircraft is a/an concept.”]
Now we think about the potential relations between these entities:
[“manufacturer”]
Let’s make a draft of the triples.
[[“Airbus A380”, “manufacturer”, “Airbus”], [“380”, “manufacturer”, “Airbus”],
[“Wide-body aircraft”, “manufacturer”, “Airbus”]]
Now it is time to think and filter out incorrect triples if there is any.
The following triples seem to be incorrect:
```

```
[["Wide-body aircraft", "manufacturer", "Airbus"], ["380", "manufacturer", "Airbus"]].
Here is the reason why I think these triples are incorrect:
["The relation - manufacturer - does not hold for Wide-body aircraft and Airbus.",
"The relation - manufacturer - does not hold for 380 and Airbus."] Therefore,
final triples should be:
[["Airbus A380", "manufacturer", "Airbus"]]
Think like a domain expert and check the validity of the triples. Keep track of
your thinking as shown in the example and extract triples from the following text.
Text: The Porsche Panamera is a mid/full-sized luxury vehicle (E-segment in Europe)
manufactured by the German automobile manufacturer Porsche. It is front-engined and
has a rear-wheel-drive layout, with all-wheel drive versions also available.
Your answer:
```

3.5. Generated knowledge prompts

Generated knowledge prompting involves leveraging a language model to create knowledge, which is then used as supplementary context when addressing a query. This methodology is noteworthy for its independence from task-specific oversight during knowledge integration and its lack of reliance on a structured knowledge base. The application of the generated knowledge prompting has been found to enhance the performance of large-scale, cutting-edge models across a variety of commonsense reasoning tasks. Notably, this approach has attained leading results on benchmarks encompassing numerical commonsense (NumerSense), general commonsense (CommonsenseQA 2.0), and scientific commonsense (QASC), indicating its effectiveness in strengthening model reasoning capabilities in the absence of traditional knowledge sources [18].

The implementation of generated knowledge prompting for KE seeks to harness the parametric knowledge embedded within LLMs regarding entities and their relationships prior to the actual extraction process. The premise underpinning this methodology is that by using LLMs parametric knowledge about entities such as their type and potential relations, the quality of the extracted knowledge triples may be improved. In our zero-shot setting, the task sequence is augmented with instructions that includes the task description, directing LLM to first generate knowledge concerning the entities mentioned in the text and any potential relations among them. This pre-generated knowledge serves as a primer for the subsequent extraction of knowledge triples.

In one-shot and few-shot experiments, examples for task demonstration are selected by the retriever. The entity type annotation from the dataset which is verbalized with the help of a template is utilized as the demonstration of generated knowledge. The following example shows the full adaptation.

```
Your task is extracting knowledge triples from text.
A knowledge triple consists of three elements: <subject - predicate - object>.
Subjects and objects are entities and the predicate is the relation between them.
Before extracting triples, generate knowledge about the entities in the text and
potential relations between them.
Here is an example:
Text: Audi AG () is a German automobile manufacturer that designs, engineers,
produces, markets and distributes luxury vehicles. Audi is a subsidiary of
Volkswagen Group and has its roots at Ingolstadt, Bavaria, Germany. Audi vehicles
are produced in nine production facilities worldwide.
Knowledge:["Automobile manufacturer is a/an
concept.", "Luxury vehicle is a/an concept."]
The following triples can be extracted considering the knowledge.
Triples: [["Audi", "Country", "German"], ["Audi", "Owned by", "Volkswagen Group"],
["Audi", "Headquarters location", "Ingolstadt"]]
The example ends here.
Generate knowledge as shown in the example and extract knowledge triples from
```

```
the text.
Text: The Porsche Panamera is a mid/full-sized luxury vehicle (E-segment in Europe)
manufactured by the German automobile manufacturer Porsche. It is front-engined and
has a rear-wheel-drive layout, with all-wheel drive versions also available.
Your answer:
```

3.6. Reason and act prompts

Combining task-oriented actions with verbal reasoning, or inner speech, is a distinct feature of human intelligence. It is believed that this capability play an important role in human cognition for enabling self-regulation or strategization and maintaining a working memory [33]. Inspired from human intelligence, Yao et al. explore the use of LLMs to generate both reasoning traces and task-specific actions in an combined manner, fostering a cooperative dynamic between the two: Reasoning and Acting, or ReAct, prompt engineering method. ReAct involves generating thoughts, action points, and observations iteratively until a task is completed, making it well-suited for interactive agents. Unlike chain of thought and chain of thought with self-consistency algorithms, which do not involve user interaction, ReAct engages users by breaking queries into intermediate steps and developing thoughts and action plans in an interactive environment. This user involvement leads to more meaningful interactions, helping agents achieve their goals and respond to user queries more satisfactorily.

Our experimental setup does not involve users, so our results do not fully capture the real potential of the ReAct method. ReAct's strength lies in its ability to generate thoughts, action points, and observations iteratively in an interactive environment, engaging users throughout the process. Without user interaction, our evaluation may not accurately reflect the method's effectiveness in achieving goals and responding to queries in a more satisfactory manner. However, we demonstrate how the ReAct method can be implemented in the context of knowledge extraction. Despite the absence of user interaction in our experimental setup, we illustrate the method's capability to generate thoughts, action points, and observations iteratively.

The ReAct prompting method is applied to KE by incorporating additional instructions. After the task description, LLM is asked to generate thoughts and make an action plan until knowledge triples are extracted. For one-shot and few-shot experiments, the execution of the task is demonstrated to LLM on the retrieved example as below.

```
Your task is extracting knowledge triples from text.
A knowledge triple consists of three elements: subject - predicate - object.
Subjects and objects are entities and the predicate is the relation between them.
Let's use an example:
Text: Audi AG () is a German automobile manufacturer that designs, engineers, pro-
duces, markets and distributes luxury vehicles. Audi is a subsidiary of Volkswagen
Group and has its roots at Ingolstadt, Bavaria, Germany. Audi vehicles are produced
in nine production facilities worldwide.
Thought 1: I need to determine the entities.
Act 1: Named entity extraction.
Observation 1: ["Audi", "German", "automobile manufacturer", "luxury vehicle",
"Volkswagen Group", "Ingolstadt", "Bavaria"]
Thought 2: What type of entities do I have?
Act 2: Named entity tagging
Observation 2: ["Automobile manufacturer is a/an concept.", "Luxury vehicle is
a/an concept."]
Thought 3: What are the potential relations between these entities?
Act3: List the potential relations
Observation3: ["country", "owned by", "headquarters location"]
Thought 4: What are the triples?
Act4: Form the triples
Observation4: [["Audi", "Country", "German"], ["Audi", "Owned by", "Volkswagen
```

```
Group"], ["Audi", "Headquarters location", "Ingolstadt"]]
Thought5: I have extracted knowledge triples from the input text.
Act5: Finish
Observation5: Task is completed.
Before answering a query, think and decide your act. Extract the knowledge triples
from the following text.
Text: The Porsche Panamera is a mid/full-sized luxury vehicle (E-segment in Europe)
manufactured by the German automobile manufacturer Porsche. It is front-engined and
has a rear-wheel-drive layout, with all-wheel drive versions also available.
Your answer:
```

4. Evaluation

Our proposed evaluation procedure is based on Wikidata and comprises two key steps. First, it is a prerequisite to establish links between the components, i.e. subject-predicate-object, of each triple and their corresponding Wikidata identifiers. Subsequently, SPARQL queries are executed to acquire semantic information pertaining to entities and relations. We scrutinize the alignment of entity types with the specified subject type restrictions (e.g. rdfs:domain) and the value type restrictions (e.g. rdfs:range) of the predicates. Conforming entity types and the domain and range are taken as a semantic confirmation of the correctness of an extracted triple. In addition to assessing correctness, the method also determines the novelty of extractions by verifying whether the extracted triples already exist within Wikidata. This comprehensive evaluation methodology not only assesses the alignment with semantics but also takes into account the novelty of the extracted triple within the broader knowledge base.

All SPARQL queries used for the evaluation can be found in Appendix A. A comprehensive example of the resulting output of the evaluation process is provided in Appendix B.

4.1. Data

We utilize RED-FM, which stands for a Filtered and Multilingual Relation Extraction Dataset, as detailed by Huguet Cabot et al., 2023 [10]. This dataset, refined through human review, encompasses 32 relations across seven different languages and is designed to facilitate the assessment of multilingual RE systems. RED-FM derives its content and structure from both Wikipedia and Wikidata [28]. The reader is referred to the original paper [10] for further details about the dataset.

For the assessment of our selected prompt engineering methods, our focus is concentrated on the human-annotated segment of the RED-FM dataset, specifically RED-FM, English. This subset contains a total of 3,770 data points. Our evaluation process of the different prompting methods utilizes the designated test split, which includes 446 instances. Furthermore, to select examples for task demonstrations for one-shot and few-shot prompts, a random selection of 300 instances has been extracted from the training section of the dataset. The chosen instances within this sample serve as exemplars to illustrate the implementation of the tasks within the prompt.

4.2. LLMs

To evaluate the effectiveness of our chosen prompt engineering strategies, we put them to the test on three different state-of-the-art LLMs: GPT-4, Mistral 7B, and Llama 3.

4.2.1. Mistral 7B

Mistral 7B [11] is a language model that has been engineered for superior performance and efficiency in NLP. It is released under the Apache 2.0 license. The model architecture is based on a transformer architecture with seven billion parameters. Mistral 7B introduces features like sliding window attention, rolling buffer cache, and pre-fill and chunking to optimize its performance. Mistral 7B has a specialized variant called Mistral 7B – Instruct, which is fine-tuned to follow instructions and outperforms several other models in benchmarks. We employ this specialized

variant[2] to test the selected prompt engineering methods. We run inference using LangChain HuggingFaceEndpoint [3] with the following parameters: Temperature 0.5, maximum number of new tokens 512.

4.2.2. Llama 3

Meta developed and released the Meta Llama 3 [1] family of large language models, comprising a collection of pretrained and instruction-tuned generative text models in 8 and 70 billion parameter sizes – referred to as 8B and 70B respectively. Llama 3 is an auto-regressive language model that uses an optimized transformer architecture. The tuned versions use supervised fine-tuning (SFT) and reinforcement learning from human feedback (RLHF) [6] to align with human preferences for helpfulness and safety. Both the 8 and 70B versions use Grouped-Query Attention (GQA) for improved inference scalability. Llama 3 is a gated model, requiring users to request access and it is released under the Meta Llama 3 Community License Agreement. For our experiments with Llama 3, we use 8B Instruct model[4] and run inference via LangChain HuggingFaceEndpoint with the following parameters: Temperature 0.5, maximum number of new tokens 512.

4.2.3. GPT-4

GPT-4 [22], the latest iteration in the Generative Pre-trained Transformer series, represents a significant advancement in large-scale, multimodal artificial intelligence models. It is important to note that GPT-4 is a proprietary technology, with access provided exclusively via an API. Regarding the performance of the model, it is reported that GPT-4 is capable of showing human-level performance on par with the top 10% of test takers in a simulated bar exam [22]. As its antecedents, it is pre-trained to predict the next token in a document using both publicly available data, i.e., internet data, and data licensed from third-party providers. The model is then finetuned using Reinforcement Learning from Human Feedback (RLHF) [6]. OpenAI (2023) [22] reports improved performance on measures of factuality and adherence to desired behavior after the post-training alignment process. For the experiments, we use OpenAI API. The temperature of the model is set to 0.5, and maximum number of tokens to 2000.

4.3. Parsing extracted triples

In this study, we observed that extracted triples can be presented in various formats, including lists of lists, lists of dictionaries, enumerated dictionaries, enumerated lists, enumerated strings with a separator (e.g., "-"), enumerated lists of tuples, JSON strings, and dictionaries of triples. The format variability necessitates a robust post-processing approach, as the outcomes may differ based on the method employed.

Our post-processing approach is designed to parse all extracted triples effectively. We operate under two key assumptions:

1. Direct Structure Assumption: We assume that the generated text inherently represents a structured format. Consequently, we attempt to load it directly. The possible structures include:
   (a) List of lists
   (b) List of dictionaries
   (c) A dictionary where each key corresponds to a list of dictionaries (triple dictionary)
   (d) JSON string
2. Mixed Content Assumption: We assume that the generated text may contain the structure intermixed with explanatory content or be presented as enumerated items. To address this, we employ regular expressions to extract the structured data before attempting to load it. This extraction process accounts for various shapes, including enumerated items such as lists, dictionaries, and tuples.

This dual-assumption approach ensures comprehensive parsing of the generated text, accommodating a wide range of formats and improving the reliability of our post-processing results.

[2] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3
[3] https://python.langchain.com/v0.2/docs/integrations/llms/huggingface_endpoint/
[4] https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct

4.4. Wikidata as reference

Wikidata operates as the structured data counterpart of Wikipedia, serving as a collaborative platform where users collectively curate and manage a knowledge graph. Knowledge within Wikidata is systematically transformed into a Semantic Web representation, ensuring that it adheres to standards suitable for integration and interoperability across various web technologies [28]. The test dataset employed in this research is sourced from Wikipedia and Wikidata, including the Wikidata identifiers of the entities and relations. While the dataset is structured to align with the specific entities and relations identified in RED-FM, inputs may semantically contain a variety of triples such as event triples, as exemplified in Fig. 1, that do not fall into RED-FM schema. This implies that the input text could potentially feature extraneous relations, not explicitly captured or categorized by the RED-FM framework. This creates an environment that includes entities and relationships outside the defined scope of the target triples in RED-FM.

The evaluation protocol used in this study involves a post-processing phase for the responses produced by the LLM in response to the prompts. As detailed in Section 4.1, these responses are parsed to identify segments that correspond to linearized triples by using Python's JSON and REGEX (regular expressions) modules. These segments are then transformed into lists that organize the data into distinct triples, each comprising a subject, a predicate, and an object. After the parsing phase, each component of the triple is validated against Wikidata through an API call, employing a greedy keyword search methodology for the purpose of entity linking. Entity linking is a prerequisite for the implementation of our evaluation framework. Upon successful linking of a triple's subject, predicate, and object to their respective Wikidata counterparts, a SPARQL "ASK" query is executed. This query serves to ascertain the existence of the triple within Wikidata. If the query returns True, it validates the LLM's output as a factual piece of knowledge. In cases that the query returns False, it indicates a novel extraction that may be added to Wikidata.

4.5. Ontology-based triple assessment

To navigate the challenges of open extraction and the expressiveness of LLMs, this study introduces an ontology-based method for assessment of the extracted triples. Ontology based triple assessment aims to reduce dependence on human evaluation. By leveraging the underlying data semantics and property restrictions present in Wikidata, it is possible to automate the validation process for the extracted triples.

The evaluation process starts by querying Wikidata for predicates to obtain their domain, identified by the *subject type constraint* (Q21503250), and range, identified by the *value-type constraint* (Q21510865). Additionally, type information for entities is extracted from Wikidata up to four hierarchical levels using the *instance of* (P31) and *subclass of* (P279) properties. The method then checks whether the predicate's domain and range align with the subject's and object's types at any level of the extracted hierarchy. This verification ensures that the subject and object types are consistent with the properties of the predicate according to Wikidata constraints. Figure 2 illustrates the assesment of the following triple: "Porsche Panamera", "manufacturer", "Porsche". The first step is checking Wikidata to find corresponding Wikidata identifiers for the triple components. In this case, all components of this triples have a Wikidata identifier: "Q501349", "P176", "Q40993". Then, we query Wikidata to check whether the predicate "P176" has a defined domain and range. "P176" is a well-defined predicate that has both domain and range restrictions.

In a triple where the predicate is "P176", the subject should be a member of one of the following classes: "software": "Q7397", "physical object": "Q223557", "model series": "Q811701", "concrete object": "Q4406616", "product model": "Q10929058". Additionally, the object of the triple should be a member of one of the following classes: "human": "Q5", "animal": "Q729", "profession": "Q28640", "organization": "Q43229", "factory": "Q83405", "fictional character": "Q95074", "industry": "Q268592", "artisan": "Q1294787", "group of fictional characters": "Q14514600". The third step is checking the subject and object types up to the fourth level in depth, then comparing those classes with the domain and range of the predicate. The subject "Porsche Panamera": "Q501349" is a "Model Series": "Q811701" that aligns with the domain of the predicate *"P176"*. The property path to "Model Series": "Q811701" is a follows: "Automobile Model Series": "Q59773381" » "Vehicle Model Series": "Q29048319" » "Model Series": "Q811701". The object "Porsche": "Q40993" is an "Organization": "Q43229" that aligns with the range of the predicate. The property path to "Organization": "Q43229" is as folows: "Racecar Constructor":

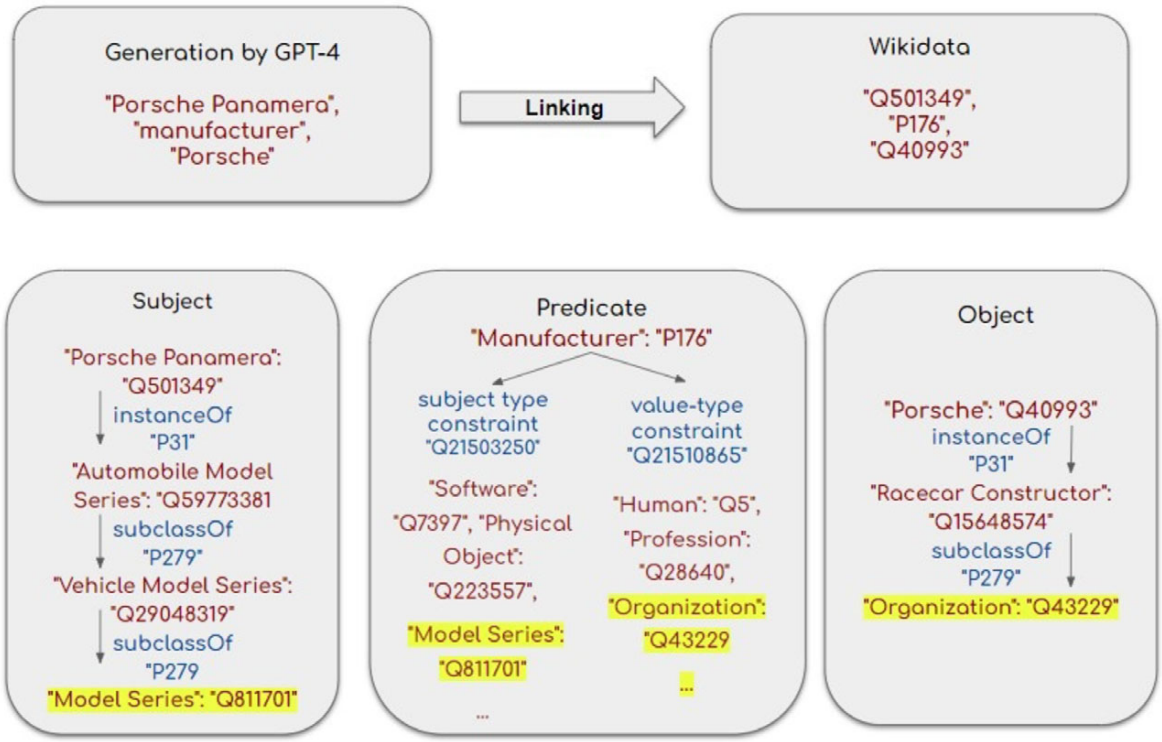

Fig. 2. An example of ontology based triple assessment.

"Q15648574" » "Organization": "Q43229". According to our ontology based evaluation framework, this extraction is deemed correct.

5. Results

This study examines the effectiveness of diverse prompting strategies applied to the RED-FM dataset, focusing on the correctness, novelty, and alignment with an ontology of extracted triples. Seventeen distinct prompt templates generated by employing five prompting methods with distinct task demonstration strategies are systematically adapted to KE and tested on Mistral 7B, Llama 3, and GPT-4 using a test set comprising 446 instances. All the applied prompting methods are listed in Table 1. The table categorizes these methods based on the prompt type as we name them, and provides details on the task demonstration approach, methodological strategy, and specific instructions used for each method. Each method contains specific instructions provided to guide LLMs in performing the KE task. These instructions outline the steps and considerations necessary for executing the task effectively, tailored to the methodology and approach of each prompting method. The table serves as a consolidated summary and comparison of the different prompting methods explored in our research. Effectiveness of each prompting method is measured through the extraction of triples. Our evaluation protocol contains following assessments:

1. Standard evaluation metrics, i.e. precision, recall, F1 score in reference to RED-FM gold standards.
2. Alignment with Wikidata ontology.
3. Additional analysis of ontological alignment such as entity and predicate linking rates in reference to Wikidata and ontological alignment level where we check domain and range matches.
4. Novelty analysis where we check whether extracted triples are novel or already in Wikidata.

The results section includes several detailed tables. The primary table, Table 2, identifies the most effective prompting method. Additionally, Table 3 presents an ontology-based assessment of the extracted triples, focusing

Table 1
Summary of tested prompting methods

Prompt Type	Demonstration	Method	Instruction
0-shot	0 task demonstration	Direct instruction	"Extract knowledge triples from the text. Return the triples in JSON format."
1-shot	1 canonical example	Direct instruction	Same as 0-shot
Few-shot	3 canonical examples	Direct instruction	Same as 0-shot
RAG 1-shot	1 retrieved example	Direct instruction	Same as 0-shot
RAG few-shot	3 retrieved examples	Direct instruction	Same as 0-shot
CoT 0-shot	0 task demonstration	Chain of Thought	"Your task is extracting knowledge triples from text. A knowledge triple consists of three elements: subject – predicate – object. Subjects and objects are entities and the predicate is the relation between them. Before extracting triples, let's think step by step."
CoT 1-shot	1 retrieved example	Chain of Thought	Same as CoT 0-shot
CoT few-shot	3 retrieved examples	Chain of Thought	Same as CoT 0-shot
Self-cons 0-shot	0 task demonstration	Chain of thought with self-consistency	"Your task is extracting knowledge triples from text. A knowledge triple consists of three elements: subject – predicate – object. Subjects and objects are entities and the predicate is the relation between them. First, think about entities and relations that you want to extract from the text. Then, look at the potential triples. Think like a domain expert and check the validity of the triples. Filter out the invalid triples. Return the valid triples in JSON format."
Self-cons 1-shot	1 retrieved example	Chain of Thought with Self-consistency	Same as Self-cons 0-shot
Self-cons few-shot	3 retrieved examples	Chain of Thought with Self-consistency	Same as Self-cons 0-shot
GenKnow 0-shot	0 task demonstration	Generated Knowledge	"Your task is extracting knowledge triples from text. A knowledge triple consists of three elements: subject – predicate – object. Subjects and objects are entities and the predicate is the relation between them. First, generate knowledge about the entities in the text and potential relations between them. Then, extract the triples. Return the triples in JSON format."
GenKnow 1-shot	1 retrieved example	Generated Knowledge	Same as GenKnow 0-shot
GenKnow few-shot	3 retrieved examples	Generated Knowledge	Same as GenKnow 0-shot
ReAct 0-shot	0 task demonstration	Reasoning and Acting	"Your task is extracting knowledge triples from text. A knowledge triple consists of three elements: subject – predicate – object. Subjects and objects are entities and the predicate is the relation between them. Generate thoughts and make an action plan for each step until you extract the knowledge triples from the following text. Return the triples in JSON format."
ReAct 1-shot	1 retrieved example	Reasoning and Acting	Same as ReAct 0-shot
ReAct few-shot	3 retrieved examples	Reasoning and Acting	Same as ReAct 0-shot

Table 2
Comparing prompting methods across LLMs

Prompt type	Mistral – 7B – Instruct			Llama 3 – 8B – Instruct			GPT-4		
	F1 Score	Domain & Range	Statement Match	F1 Score	Domain & Range	Statement Match	F1 Score	Domain & Range	Statement Match
0-shot	0.00	0.43	0.14	0.00	0.24	0.07	0.00	0.48	0.15
1-shot	0.04	0.48	0.18	0.03	0.44	0.14	0.02	0.53	0.14
Few-shot	0.04	0.50	0.18	0.03	**0.52**	0.19	0.03	0.56	0.16
RAG 1-shot	0.09	0.56	0.25	0.08	0.48	0.17	0.07	0.60	0.27
RAG few-shot	**0.10**	**0.61**	**0.28**	**0.10**	0.47	0.19	**0.11**	**0.61**	**0.29**
CoT 0-shot	0.00	0.36	0.15	0.00	0.29	0.07	0.00	0.34	0.09
CoT 1-shot	0.06	0.50	0.20	0.07	0.47	0.18	0.05	0.54	0.21
CoT few-shot	0.06	0.49	0.19	0.05	0.46	0.16	0.06	0.50	0.22
Self-cons 0-shot	0.00	0.36	0.12	0.00	0.24	0.06	0.00	0.44	0.15
Self-cons 1-shot	0.06	0.39	0.17	0.07	0.41	0.16	0.04	0.46	0.18
Self-cons few-shot	0.07	0.46	0.18	0.03	0.44	0.12	0.06	0.51	0.18
GenKnow 0-shot	0.00	0.39	0.12	0.00	0.22	0.05	0.00	0.46	0.12
GenKnow 1-shot	0.07	0.54	0.22	0.05	0.43	0.16	0.04	0.54	0.23
GenKnow few-shot	0.07	0.53	0.22	0.06	**0.52**	**0.20**	0.09	**0.61**	0.26
ReAct 0-shot	0.00	0.47	0.11	0.00	0.32	0.08	0.00	0.46	0.12
ReAct 1-shot	0.04	0.43	0.14	0.04	0.45	0.17	0.04	0.50	0.19
ReAct few-shot	0.06	0.48	0.21	0.04	0.40	0.14	0.07	0.50	0.23

on the performance of Mistral 7B while the assessments for GPT-4 and Llama 3 are available in the Appendix C. Table 5 offers a comprehensive novelty analysis of the extracted triples for GPT-4 with similar analyses for the other two models found in the Appendix C. There is also a table detailing the analysis of extracted entities and relations for Llama 3 with corresponding details for the other two models in the Appendix C. Furthermore, the section includes Fig. 3 analyzing the ontological alignment levels, with supplementary data for the other two models provided in the Appendix C. Finally, a summary chart, Fig. 4 encapsulates all analyses for the best performing prompting method.

5.1. Comparing prompting methods across LLMs

Table 2 encapsulates the major outcomes of our experiments where we try to answer which prompt engineering method performs the best for KE from text. The first column denotes the abbreviation of the applied prompting strategy. For each of the three LLMs, the table reports the performance metrics for all the prompt engineering methods summarized in Table 1. The highest scores in each column are highlighted in bold font. The table includes the following metrics for each prompting method: F1 Score, Domain & Range Match, and Statement Match.

F1 Score: This metric is calculated in reference to RED-FM target triples.
Domain & Range Match: This column indicates the percentage of extracted triples that have a matching domain and range according to our ontology-based assessment.
Statement Match: This column represents the percentage of triples that are already present in Wikidata as statements.

Key observations from the table reveal that all tested LLMs perform poorly on the task when evaluated against the target triples annotated in the RED-FM test set, with the highest F1 score recorded at 0.11. However, the F1 score alone does not sufficiently indicate the quality of the extraction because of the language variation comes with generative language models in open information extraction setting as discussed in detail in previous sections and illustrated in Fig. 1. The Domain & Range Match and Statement Match metrics indicate the ontological alignment of the extraction with reference to Wikidata, providing insights into the quality and correctness of the extraction. These two metrics display more variation than the F1 score, offering deeper insights into the efficacy of the implemented prompting strategy.

Table 3
Ontology based triple assessment results of Mistral 7B

Prompt type	Number of extracted triples	Triples with a well-defined predicate		Triples with a well-defined predicate and both entity types known		Domain & Range matching triples	
		Nr	%	Nr	%	Nr	%
0-shot	2284	400	0.18	182	0.46	78	0.43
1-shot	5047	1300	0.26	941	0.72	454	0.48
Few-shot	4018	1162	0.29	853	0.73	428	0.50
RAG 1-shot	2839	924	0.33	727	0.79	406	0.56
RAG few-shot	2498	813	0.33	656	0.81	398	0.61
CoT 0-shot	3078	432	0.17	185	0.43	67	0.36
CoT 1-shot	3739	972	0.26	727	0.75	367	0.50
CoT few-shot	3532	952	0.27	672	0.71	331	0.49
Self-cons 0-shot	3022	501	0.18	276	0.55	99	0.36
Self-cons 1-shot	4801	1406	0.29	1043	0.74	405	0.39
Self-cons few-shot	4231	1224	0.29	924	0.75	422	0.46
GenKnow 0-shot	2696	493	0.18	267	0.54	103	0.39
GenKnow 1-shot	3633	979	0.27	735	0.75	396	0.54
GenKnow few-shot	3531	971	0.30	741	0.76	392	0.53
ReAct 0-shot	2223	282	0.18	120	0.43	56	0.47
ReAct 1-shot	3684	918	0.25	605	0.66	263	0.43
ReAct few-shot	3599	822	0.27	597	0.73	287	0.48

Incorporating task demonstrations consistently improves performance, particularly for the Domain & Range Match and Statement Match metrics. The best performing prompt is task demonstrations selected by a retrieval mechanism (RAG few-shot), which achieves the highest F1 scores and Statement Match rates across a majority of models, indicating superior extraction with this method. Notably, Domain & Range Match scores show an improvement of +0.5 for Mistral 7B and +0.1 for GPT-4, although Llama 3's performance on this metric is negatively impacted by the addition of retrieved examples. Llama 3 seems to benefit more from canonical examples as it is reflected on Domain & Range, and Statement Match metrics. The Self-Consistency and Generated Knowledge methods show significant improvements across all metrics when task demonstrations are incorporated into the prompts, particularly Llama 3. However, these methods exhibit moderate improvements or some degradation in few-shot scenarios, as seen in the performance of Mistral 7B with Generated Knowledge or Llama 3 with Reasoning and Acting prompts in the Domain & Range Match metric. Finally, Reasoning and Acting prompting demonstrate competitive performance with Chain of Thought or Self consistency methods in one-shot and few-shot approaches compared to zero-shot scenarios.

5.2. Ontological alignment

Ontologies serve as the backbone of knowledge representation within knowledge graphs, providing a structured framework that delineates the classes, properties, and interrelationships of the included information [8]. Taking this into account, an extensive analysis of the extracted triples, entities and relations is undertaken. The methodology employed harnesses the hierarchical structure of Wikidata, examining entity types up to the fourth level in depth. Domain and range specifications of relations are also taken into account to ascertain their designated function within the knowledge graph.

The findings, detailed in Table 3, encapsulate the major outcomes of our ontology-based assessment for the extraction performed by Mistral 7B. Results for the other two LLMs can be found in Appendix C as in Table 6 and Table 7. These models demonstrate comparable trends. Notably, both GPT-4 and Mistral achieve a Domain

& Range matching rate of 0.61. Upon examination of the aggregate results, GPT-4 exhibits a marginally superior performance. However, for the purpose of demonstrating overall ontology-based triple assessment results, we have opted to focus on Mistral 7B.

The initial column in Table 3, *Number of extracted triples* enumerates the total count of extracted triples by Mistral 7B for each prompting strategy, computed across the 446 test instances. The second column *Triples with a well-defined predicate* denotes the quantity of triples linked to Wikidata and featuring a well-defined predicate, i.e. rdfs:domain and rdfs:range. The third column articulates the proportion of such triples relative to the overall extracted set. Additionally, the table includes further details to provide a comprehensive overview. The fourth and fifth columns *Triples with a well-defined predicate and both entity types known* detail the number of triples with a well-defined predicate and both subject and object are also linked to Wikidata, as well as the percentage relative to the total triples with a well-defined predicate. The last two columns *Domain and Range matching triples* provide insights into the alignment between the well-defined predicates and the types of entities which form the triple under assessment. Specifically, the seventh column details the number of triples where the type of the subject matches the domain of the predicate, and the range of the predicate aligns with the type of the object. The last column presents the percentage of such domain and range-matched triples relative to all triples with a well-defined predicate. These statistics offer a refined evaluation of the semantic coherence and structural alignment within the extracted triples.

Analyzing Table 3 consolidates our analyses in Table 2. Consistent lowest performance of zero-shot prompts emphasizes the challenges in generating accurate extractions without specific task demonstrations. Zero-shot prompts are designed to consist solely of a brief directive, instructing the model to extract knowledge triples from the text. Interestingly, empirical findings indicate that this minimalist prompting approach yields performance comparable to that of the Self-consistency method, which employs a more intricate strategy of generating multiple outputs and selecting the most consistent answer. Moreover, this direct zero-shot technique demonstrates a performance advantage, outpacing other zero-shot methods by 4-6% in Domain & Range Match. These alternative methods, such as Chain of Thought (CoT), Self consistency (Self-cons), and Generated Knowledge (GenKnow) incorporate additional context or steps to guide the language model. The effectiveness of the straightforward instruction employed in the zero-shot prompt underscores the capability of advanced language models to execute tasks with minimal instructions.

In-context Learning (ICL) leverages the concept of learning by demonstration. This phenomenon enables LLMs to adjust their generated responses to align with the demonstrated examples, emulating the showcased task-specific behavior [14]. Our experimental results supports the phenomena since the introduction of task demonstrations into the prompts significantly enhances performance, with a notable increase, i.e. from 36% (CoT 0-shot) up to 50% (CoT 1-shot), compared to the zero-shot approaches as shown in Table 3. Incorporating a single canonical example notably elevates the model's performance, evidenced by an increase from a baseline of 0.43 to an enhanced measure of 0.48. However, incorporating an additional pair of canonical examples within a few-shot learning scenario yields a marginal enhancement in model performance, indicated by a mere 2% increment. The marginal impact of additional examples suggests a saturation point in the model's ability to benefit from increased prompt complexity, in some cases adding more examples negatively impacts the performance as reflected on the results of Chain of Thought and Generated Knowledge prompts.

The selection of demonstration examples in ICL is important due to several reasons that influence the performance of the model. The examples prime the model with an indication of what the task entails. Following the incorporation of examples retrieved from the training sample, observed decrease in the number of extracted triples, e.g, from 2839 to 2498 in Table 3 while F1 score is increasing in RAG few-shot prompts may suggest that the selected examples are influencing the extraction process. This could potentially indicate a refinement in the model's focus or an alignment of its extraction criteria more closely with the characteristics of the provided examples, consequently impacting the volume of extracted triples. The reported decrease in the total amount of extracted triples, coupled with the observed increase in triples conforming to the Wikidata ontology (Domain&Range match), implies a qualitative enhancement in the extraction method subsequent to the incorporation of the retrieved examples.

Intriguingly, the adaption of prompting methods such as Chain-of-Thought (CoT), Self-consistency, and Reasoning and Acting (ReAct), devised to improve model's reasoning aptitude does not appear to yield any substantial enhancements in the performance of open knowledge triple extraction. This pattern intimates that the integration of reasoning-focused cues within the prompts does not markedly improve the model's proficiency in formulating

Table 4
Analysis of entities and relations extracted by Llama 3

Prompt type	Entity Analysis			Relation Analysis				
	All	Linked		All	Linked		has Domain&Range	
	Nr	Nr	%	Nr	Nr	%	Nr	%
0-shot	3815	1824	0.48	1604	479	0.30	192	0.40
1-shot	4123	2368	0.57	1562	538	0.34	226	0.42
Few-shot	4582	2639	0.58	1728	482	0.28	210	0.44
RAG 1-shot	4311	3185	0.74	1303	669	0.51	298	0.45
RAG few-shot	3598	2809	0.78	1043	577	0.55	243	0.42
CoT 0-shot	8865	2572	0.29	1574	496	0.32	228	0.46
CoT 1-shot	4176	2746	0.66	1627	668	0.41	305	0.46
CoT few-shot	4307	306	0.70	1579	647	0.41	287	0.44
Self-cons 0-shot	6191	2496	0.40	2030	569	0.28	247	0.43
Self-cons 1-shot	4733	2666	0.56	1991	610	0.31	272	0.45
Self-cons few-shot	4189	2183	0.52	1770	526	0.30	242	0.46
GenKnow 0-shot	4521	2369	0.52	1806	559	0.31	240	0.43
GenKnow 1-shot	3464	2266	0.65	1318	555	0.42	252	0.45
GenKnow few-shot	3839	2649	0.69	1296	546	0.42	242	0.44
ReAct 0-shot	2339	1260	0.54	1090	323	0.30	134	0.41
ReAct 1-shot	4240	2269	0.54	1844	657	0.36	298	0.45
ReAct few-shot	4693	2557	0.54	2025	652	0.32	293	0.45

correct responses or tackling intricate problems within the examined experimental framework. However, it can be argued that reasoning about entity types may assist humans in extracting more accurate triples from text.

When examining the efficacy of these three reasoning-oriented prompting methods individually, it is noted that their performance metrics cluster within a similar range. Across the methods tested, there is no recognizable performance gap substantial enough to distinguish any single method as being superior. This lack of significant differentiation in performance further reinforces the findings that, within the experimental parameters set for open knowledge triple extraction, reasoning-enhanced prompting does not markedly benefit the model's output.

5.2.1. Analysis of extracted entities and relations

The evaluation process for the extracted triples is anchored in the semantics delineated by the Wikidata ontology. A focus on semantic integrity is prevalent for confirming the ontological agreement of the extracted data with the structure of Wikidata. Additionally, we postulate that looking into unique extractions is indicative to understand the quality of the extraction and the efficacy of the implemented prompting strategy. In addition to the results presented in Table 3, we look into the entities and relations extracted through studied prompting strategies in Table 4, that is offering a deeper perspective on the effect of applied prompt engineering method on the performance of the extraction regarding the alignment with Wikidata.

The entity analysis section in Table 4 displays the total number of unique entities extracted by Llama 3, those successfully linked to Wikidata, and the percentage of linked entities relative to the overall extractions. Analysis of entities and relations extracted by GPT-4 and Mistral 7B can found in Appendix C as they are presented in Table 10 and Table 11. These two models demonstrate trends similar to Llama 3. However, for the purpose of displaying overall entity and relation linking analysis, we have opted to focus on Llama 3 due to its superior performance in relation linking rate. The relation analysis segment of Table 4 shows the total count of unique relations extracted, the subset successfully linked to Wikidata, and the corresponding percentages. Moreover, it assesses the presence of domain and range restrictions within the linked relations, providing a deeper understanding of the availability of data semantics.

Table 4 provides insights into the linking rates of entities and relations. Compared to relations, entities demonstrate higher linking rates in all prompting methods. Linking rate reaches up to 78% for the RAG prompts in the

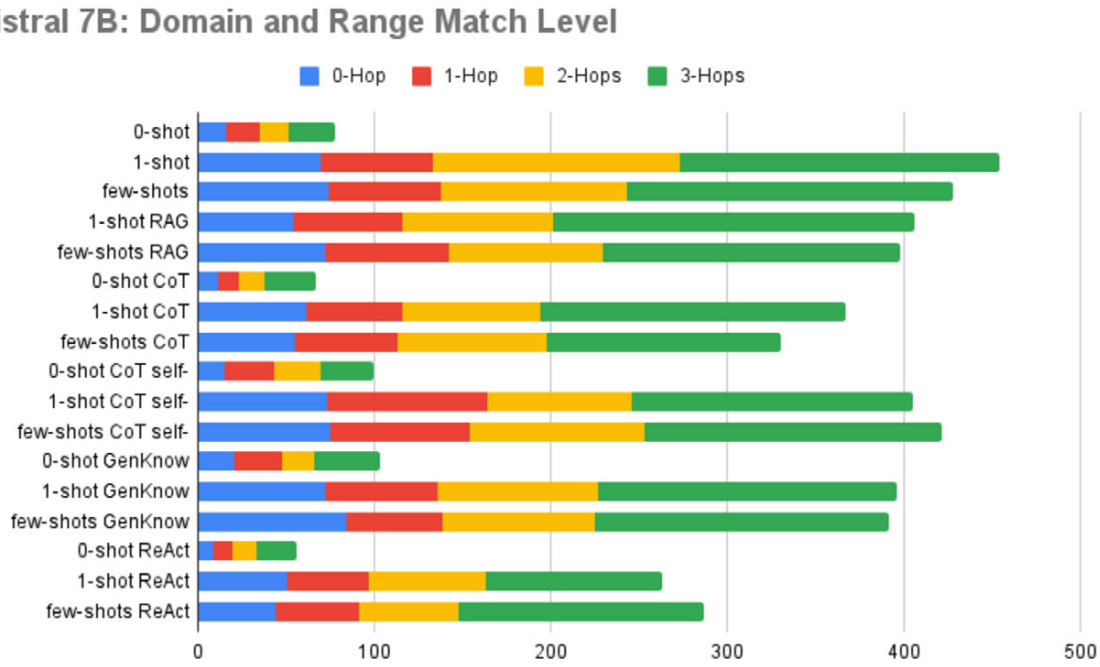

Fig. 3. Analysis of domain & range matching levels of Mistral 7B.

few-shots setting. Relations, however, exhibit a significantly lower linking rate, capped at 55% again for the RAG prompts in the few-shots setting. Furthermore, the statistics show that only 40 to 46% of linked predicates are actually well-defined in Wikidata, highlighting potential areas for improvement.

5.2.2. Matching level

Another aspect that we investigate for a deeper understanding of ontological alignment of extractions is the Domain & Range matching level. The studies presented in Fig. 3 detail the distribution of domain and range matching levels for each prompting method. We have opted to focus on Mistral 7B in this section to ensure consistency and facilitate easier comparison, as we previously showed the Domain & Range match rate in Table 3. The Domain & Range matching level analyses for the other two models exhibit similar trends and can be found in Appendix C, specifically in Fig. 5 and Fig. 6.

The matching levels include 0-Hop (direct match), 1-Hop (match one level above), 2-Hop (match two levels above), and 3-Hop (match three levels above). These levels provide insights into the depth of semantic alignment between entities and predicates. The majority of matches occurs on the upper levels, particularly on the 3-hops that corresponds to the 4th level, underscores the necessity of using different levels of granularity and hierarchy in an ontology.

5.3. Novelty of the extraction

Our empirical results suggest that LLMs possess the capability to mine novel information in the form of knowledge triples from Wikipedia text, which align with the existing schema of the Wikidata ontology. These novel extractions represent the potential for improving knowledge graphs in terms of comprehensiveness. The novelty evaluation, presented in Table 5, also provides a comprehensive overview of the extracted triples by GPT-4 and their alignment with Wikidata. The other two novelty evaluations show similar trends and can found in Appendix C as they are presented in Table 8 and Table 9. We have chosen to present the results of GPT-4 in this section because it yields the highest percentage of triples (26%) that are already in Wikidata, with Mistral 7B closely following with a mere 1% difference.

The structure of Table 5 follows a similar pattern with Table 3. *Triples linked to Wikidata* columns highlights the number of triples with all components linked to Wikidata, and the percentage compared to the total extractions.

Table 5
Assessing novelty of the extraction of GPT-4

Prompt type	Triples linked to Wikidata		Triples already in Wikidata		Novel Triples	
	Nr	%	Nr	%	Nr	%
0-shot	470	0.13	72	0.15	398	0.85
1-shot	1273	0.33	183	0.14	1090	0.86
Few-shot	1150	0.40	188	0.16	962	0.84
RAG 0-shot	1357	0.40	361	0.27	996	0.73
RAG few-shot	1515	0.52	436	0.29	1079	0.71
CoT 0-shot	420	0.11	37	0.09	383	0.91
CoT 1-shot	1420	0.42	293	0.21	1127	0.79
CoT few-shot	1324	0.49	293	0.22	1031	0.78
Self-cons 0-shot	508	0.15	75	0.15	433	0.85
Self-cons 1-shot	1510	0.45	279	0.18	1231	0.82
Self-cons few-shot	1536	0.54	281	0.18	1255	0.82
GenKnow 0-shot	426	0.12	51	0.12	375	0.88
GenKnow 1-shot	1322	0.40	306	0.23	1016	0.77
GenKnow few-shot	1462	0.49	373	0.26	1089	0.74
ReAct 0-shot	432	0.14	53	0.12	379	0.87
ReAct 1-shot	1290	0.44	251	0.19	1039	0.81
ReAct few-shot	1208	0.52	273	0.23	935	0.77

Triples already in Wikidata columns provides the count and percentage of the triples that already exist in Wikidata as statements, obtained through SPARQL "ASK" queries, this metric is also referred as Stament Match in Table 2. The central point of interest is the last two columns *Novel Triples*, showcasing the number of extracted triples whose components are all linked to Wikidata but are not present in Wikidata as a statement. This number reflects the LLM's capability to extract novel triples that could contribute to completion of knowledge graphs like Wikidata. These novel triples represent potential additions to the existing knowledge graphs.

Examining Table 5, it becomes evident that the LLM demonstrates the capacity to extract a substantial amount of triples that can easily be linked to Wikidata. A significant proportion of the extracted triples, i.e. ranging between 40% to 54% in few-shot settings, can be easily linked to Wikidata. It is particularly notable that within this subset of triples, maximum 29% is already present in Wikidata. The novel extractions present an opportunity to expand the knowledge graph. From the extraction quality perspective, it can also be argued that a higher Statement Match (Triples already in Wikidata) score may indicate superior extraction capabilities. This is because a higher Statement Match score suggests that the model can generate triples that closely resemble those already present in Wikidata. The ability to produce such similar triples implies that the model effectively captures the structure and semantic relationships inherent in the data. Consequently, a higher Statement Match score reflects the model's proficiency in accurately replicating the factual information found in a reliable and comprehensive knowledge base like Wikidata. This metric, therefore, may serve as an indicator of the model's overall extraction quality and its effectiveness in generating precise and contextually relevant information.

5.4. Reviewing the best performing prompt method

Among the all tested prompt engineering methods and LLMs, RAG (Retrieval Augmented Generation) few-shot emerges as the most effective strategy, achieving an F1 score of 0.11 by GPT-4, closely followed by Mistral 7B and Llama 3 with an F1 score of 0.10. Precision and recall scores can be found in Appendix C as they are presented in Table 12.

Figure 4 summarizes all evaluation metrics obtained by employing RAG few-shot prompting on Mistral 7B, Llama 3, and GPT-4. Each metric reflects a different aspect of the models' performance in the task of knowledge triple extraction. Linked Entities measures the proportion of entities linked to an entry in Wikidata. Mistral achieves a

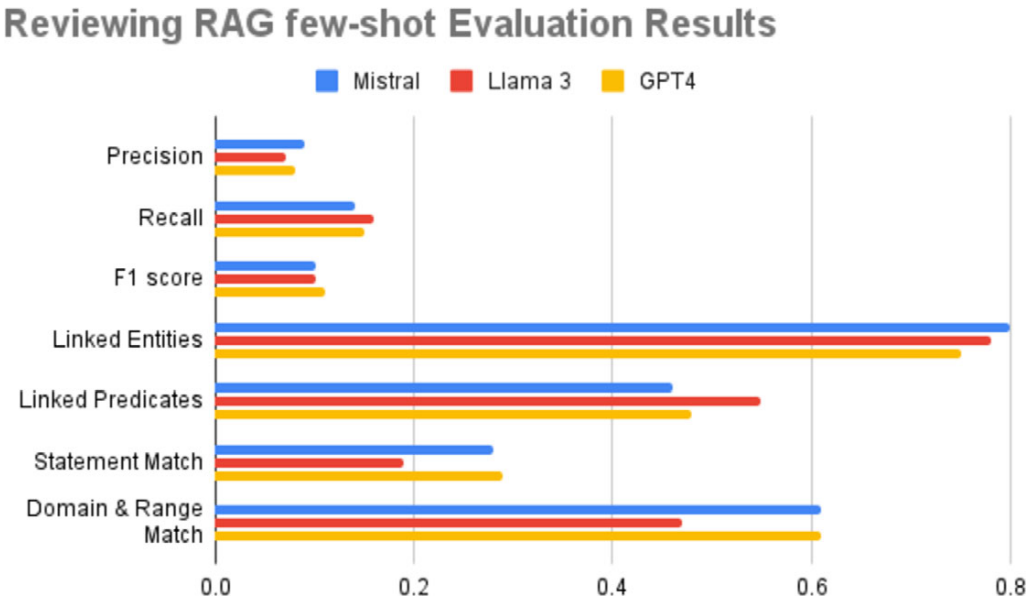

Fig. 4. Reviewing RAG few-shot evaluation results.

score of 0.80, Llama 3 scores 0.78, and GPT-4 scores 0.75. Linked Predicates measures the proportion of predicates linked to entries in Wikidata. Mistral scores 0.46, Llama 3 scores 0.55, and GPT-4 scores 0.48. Statement Match measures how well the extracted triples match the statements in Wikidata. Mistral has a statement match score of 0.28, Llama 3 scores 0.19, and GPT-4 scores 0.29. Domain & Range Match measures how well the extracted triples match the expected domain and range of the predicates. Mistral and GPT-4 both score 0.61, while Llama 3 scores 0.47. In summary, GPT-4 generally performs better in terms of F1 score and Statement Match, while Mistral performs better in precision, Linked Entities and Domain & Range Match. Llama 3 shows relatively higher recall and Linked Predicates.

6. Discussion

Our investigation into prompt engineering methods for knowledge extraction from text has yielded several salient outcomes. First, it is feasible to evaluate extracted triples in reference to a knowledge graph which has an ontology such as Wikidata. A prerequisite for this evaluation is entity and relation linking, and a simple keyword-search-based linking approach proved effective, allowing us to connect up to 80% of the extracted entities and 55% of relations to Wikidata. This linking process also revealed that, although half of the linked relations are well-defined within Wikidata, a significant portion lacks precise specifications.

Despite the feasibility of this type of evaluation, there are some concerns that we must take into account. The Wikidata ontology is dynamic, and minor user edits in top-level classes or metaclasses could significantly affect instance data. Therefore, we perform the complete evaluation within a short, well-documented time frame to ensure that potential ontology edits during the evaluation process have minimal or no effect on the results. Additionally, using Wikidata as a ground truth has certain shortcomings because the Wikidata ontology can be quite noisy and prone to various types of errors [2]. Morover, the entity and property linking performed in this work is relatively rudimentary and has several limitations such as linking errors. Our method sometimes fails to find a match when one exists (false negatives) or links to the wrong entity or relation (false positives). These errors can significantly affect the evaluation, leading to inaccuracies in the assessment of our knowledge extraction methods. To address these

limitations, employing more sophisticated linking algorithms could improve the accuracy of entity and property matches. Techniques like embedding-based entity linking, which considers the context of entities, could reduce false positives and negatives. Regardless of those shortcomings, our study successfully demonstrates the potential of this evaluation approach.

Our second finding is that LLMs are capable of extracting up to 61 triples per instance, suggesting substantial extractive capability. Notably, the addition of a single task demonstration to the prompts elevates extraction performance significantly, with diminishing returns yielded by subsequent examples. Interestingly, while 26% (max) of the extracted triples has preexisting entries in Wikidata, the majority represents novel extractions. Third, among the various prompting strategies assessed, reasoning-focused methods did not show improvement over simpler prompt engineering methods. Retrieval augmentation, however, improved the model's ability to accurately extract triples without necessitating complex instructions.

RED-FM constitutes a dataset designed for the purpose of relation extraction, centering on 32 discrete relations. For this study, this dataset is utilized in the context of open knowledge extraction task. The scores obtained by Mistral 7B, Llama 3 and GPT-4 are significantly lower than the fully supervised models, i.e. mREBEL and its variants, presented in [10]. In the best case scenario, GPT-4 reaches up to 0.11 F1 score while mREBEL [10] achieves 0.54 F1 score on the same English RED-FM test set. Additionally, it is important to note that Mistral 7B, Llama 3, and GPT-4 differ in size. Mistral has 7 billion parameters, while the version of Llama 3 used in this study has 8 billion parameters. GPT-4 is significantly larger than these open-source models and it has 1.76 trillion parameters. The size of a LLM indicates its capacity to store information in its parameters, which can influence its performance in tasks such as knowledge extraction.

Finally, the analysis of the studied prompt engineering methods uncovered their relative effectiveness in the open extraction setting. This examination sheds light on the promise of LLMs in augmenting Knowledge Graphs while highlighting the intricacies involved in aligning model outputs with external knowledge bases. These insights suggest that the translation of unstructured text into structured knowledge may be more akin to a format conversion task rather than a reasoning task. Hence, conceptualizing the extraction process as a reasoning exercise may not align with the task's intrinsic nature or LLMs' inner working. Rather, the outcomes suggest an interpretation of extraction as a transformation from unstructured to structured format, underscoring the role of well-formulated examples in optimizing this "translation".

7. Conclusion

In this study, we investigated the efficacy of state-of-the-art prompt engineering methods for the KE task. We evaluate using both standard metrics as well as using a new evaluation protocol based on Wikidata. Our results show that few-shot RAG prompts perform best across multiple LLMs. Future work would benefit from testing the approach against a diverse array of datasets, utilizing varying linking mechanisms. Such an expansion would robustly ascertain the generalizability and efficacy of the ontology-based evaluation framework introduced here, refining the strategies for knowledge extraction and graph enrichment across broader and more varied domains.

Acknowledgements

This work is funded by the European Union's Horizon Europe research and innovation programme within the ENEXA project (grant Agreement no. 101070305).

Sincere appreciation goes out to Prompt Engineering Guide [24] for providing valuable insights and inspiration for selecting and adapting prompt engineering methods for this work.

Appendix A. SPARQL queries

A.1. Domain query template

```
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX p: <http://www.wikidata.org/prop/>
PREFIX ps: <http://www.wikidata.org/prop/statement/>

SELECT ?domain ?domainLabel
WHERE {

  wd:$item p:P2302 [ps:P2302 wd:Q21503250; pq:P2308 ?domain].

  SERVICE wikibase:label { bd:serviceParam wikibase:
  language "[AUTO_LANGUAGE], en". }
  }
```

A.2. Range query template

```
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX p: <http://www.wikidata.org/prop/>
PREFIX ps: <http://www.wikidata.org/prop/statement/>

SELECT ?range ?rangeLabel
WHERE {

  wd:$item p:P2302 [ps:P2302 wd:Q21510865; pq:P2308 ?range].

  SERVICE wikibase:label { bd:serviceParam wikibase:
  language "[AUTO_LANGUAGE], en". }
  }
```

A.3. Entity type query templates

A.3.1. Instance of

```
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>
PREFIX bd: <http://www.bigdata.com/rdf#>
PREFIX wikibase: <http://wikiba.se/ontology#>

SELECT ?instanceOf ?instanceOfLabel ?superClass ?superClassLabel
  ?superSuperClass ?superSuperClassLabel ?superSuperSuperClass
  ?superSuperSuperClassLabel
WHERE { wd:$item wdt:P31 ?instanceOf.
  ?instanceOf wdt:P279 ?superClass.
  ?superClass wdt:P279 ?superSuperClass.
  ?superSuperClass wdt:P279 ?superSuperSuperClass.

  SERVICE wikibase:label { bd:serviceParam wikibase:language
  "[AUTO_LANGUAGE], en". }
  }
```

A.3.2. Subclass of

```
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>
PREFIX bd: <http://www.bigdata.com/rdf#>
PREFIX wikibase: <http://wikiba.se/ontology#>

SELECT ?subclassOf ?subclassOfLabel ?superClass ?superClassLabel
  ?superSuperClass ?superSuperClassLabel ?superSuperSuperClass
  ?superSuperSuperClassLabel
WHERE { wd:$item wdt:P279 ?subclassOf.
  ?instanceOf wdt:P279 ?superClass.
  ?superClass wdt:P279 ?superSuperClass.
  ?superSuperClass wdt:P279 ?superSuperSuperClass.

  SERVICE wikibase:label { bd:serviceParam wikibase:language
  "[AUTO_LANGUAGE], en". }
  }
```

A.4. Statement checking query template

```
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX p: <http://www.wikidata.org/prop/>
PREFIX ps: <http://www.wikidata.org/prop/statement/>

ASK WHERE {{
  wd:{subject} p:{predicate} ?statement.
  ?statement ps:{predicate} wd:{object}.
  }}
```

Appendix B. A comprehensive example of conducted analysis on an extracted triple

```
"porsche panamera && manufacturer && porsche": {
"Extracted from": "The Porsche Panamera is a mid/full-sized luxury
vehicle (E-segment in Europe) manufactured by the German automobile
manufacturer Porsche. It is front-engined and has a rear-wheel-drive
layout, with all-wheel drive versions also available.",
"String triple": [
  "porsche panamera",
  "manufacturer",
  "porsche"
  ],

"Wiki triple": [
  "Q501349",
  "P176",
  "Q40993"
  ],

"has Domain": true,
```

```
"Domain": {
  "software": "Q7397",
  "physical object": "Q223557",
  "model series": "Q811701",
  "concrete object": "Q4406616",
  "product model": "Q10929058"
  },
"has Range": true,
"Range": {
  "human": "Q5",
  "animal": "Q729",
  "profession": "Q28640",
  "organization": "Q43229",
  "factory": "Q83405",
  "fictional character": "Q95074",
  "industry": "Q268592",
  "artisan": "Q1294787",
  "group of fictional characters": "Q14514600"
  },

"Predicate has Domain & Range": true,
"Subject instanceOf": {
  "zero_hop": {
    "automobile model series": "Q59773381"
    },
   "one_hop": {

    "vehicle model series": "Q29048319"
    },
  "two_hop": {
    "model series": "Q811701"
    },
  "three_hop": {
    "series": "Q20937557"
    },
  }
"Subject subclassOf": {
  "zero_hop": {},
   "one_hop": {},
  "two_hop": {},
  "three_hop": {},
  },
"Domain match": true,
"Domain matching level": 2,
"Domain matched label": "model series",
"Domain matched wikiID": "Q811701",
"Object instanceOf": {
  "zero_hop": {
    "public company": "Q891723",
    "automobile manufacturer": "Q786820",
    "racecar constructor": "Q15648574",
```

```
    "car brand": "Q10429667"
    },
  "one_hop": {
  "joint-stock company": "Q134161",
  "enterprise": "Q6881511",
  "organization": "Q43229",
  "automobile manufacturer": "Q786820",
  "manufacturer": "Q13235160",
  "brand": "Q431289"
   },

"two_hop": {
  "limited company": "Q33685",
  "juridical person": "Q155076",
  "operation": "Q362482",
  "social system": "Q1639378",
  "business": "Q4830453",
  "enterprise": "Q6881511",
  "manufacturer": "Q13235160",
  "provider": "Q13420330",
  "group of humans": "Q16334295",
  "artificial object": "Q16686448",
  "class": "Q16889133",
  "person or organization": "Q106559804"
  },

"three_hop": {
  "commercial company": "Q567521",
  "organization": "Q43229",
  "legal person": "Q3778211",
  "goods": "Q28877",
  "organizational unit": "Q679206",
  "economic agent": "Q1415187",
  "system": "Q58778",
  "juridical person": "Q155076",
  "economic entity": "Q12569864",
  "operation": "Q362482",
  "business": "Q4830453",
  "provider": "Q13420330",
  "group of living things": "Q16334298",
  "object": "Q488383",
  "abstract entity": "Q7048977",
  "collective entity": "Q99527517",
  "agent": "Q24229398"
  }

},

"Object subclassOf": {
  "zero_hop": {},
   "one_hop": {},
```

```
    "two_hop": {},
    "three_hop": {},
    },

  "Range match": true,
  "Range matching level": 1,
  "Range matched label": "organization",
  "Range matched wikiID": "Q43229",
  "All components in Wikidata": true,
  "Triple statement in Wikidata": true,
  "Domain&Range Match": true,
  "Triple matching level": 2,
  "Both Statement and Domain&Range Match": true
  },
```


Appendix C. Supplementary material

Table 6
Ontology based triple assessment results of GPT-4

Prompt type	Number of extracted triples	Triples with a well-defined predicate		Triples with a well-defined predicate and both entity types known		Domain & Range matching triples	
		Nr	%	Nr	%	Nr	%
Zero-shot	3506	300	0.09	221	0.74	107	0.48
One-shot	3806	924	0.24	626	0.68	330	0.53
Few-shot	2871	768	0.27	550	0.72	308	0.54
RAG one-shot	3414	955	0.28	692	0.72	413	0.60
RAG few-shot	2939	1021	0.35	774	0.76	474	0.61
CoT zero-shot	3951	418	0.11	160	0.38	54	0.34
CoT one-shot	3348	1024	0.31	737	0.72	395	0.54
CoT few-shot	2710	974	0.36	740	0.76	373	0.50
Self-cons zero-shot	3439	576	0.17	229	0.40	100	0.44
Self-cons one-shot	3353	984	0.29	746	0.76	346	0.46
Self-cons few-shot	2854	974	0.34	775	0.80	392	0.51
GenKnow zero-shot	3448	541	0.16	181	0.33	83	0.46
GenKnow one-shot	3292	952	0.29	694	0.73	378	0.54
GenKnow few-shot	2996	960	0.32	736	0.77	452	0.61
ReAct zero-shot	3182	508	0.16	199	0.39	92	0.46
ReAct one-shot	2910	931	0.32	703	0.76	354	0.50
ReAct few-shot	2329	861	0.37	672	0.78	339	0.50

Table 7
Ontology based triple assessment results of Llama 3

Prompt type	Number of extracted triples	Triples with a well-defined predicate		Triples with a well-defined predicate and both entity types known		Domain & Range matching triples	
		Nr	%	Nr	%	Nr	%
0-shot	4310	706	0.18	388	0.55	92	0.24
1-shot	3934	940	0.24	670	0.71	298	0.44
Few-shot	5067	1201	0.24	879	0.73	453	0.52
RAG 1-shot	5655	1611	0.28	1253	0.78	607	0.48
RAG few-shot	5202	1651	0.32	1366	0.83	641	0.47
CoT 0-shot	9644	1504	0.17	273	0.18	80	0.29
CoT 1-shot	8654	2233	0.26	1765	0.79	838	0.47
CoT few-shot	7568	2266	0.30	1872	0.83	864	0.46
Self-cons 0-shot	6510	1147	0.18	574	0.50	139	0.24
Self-cons 1-shot	7040	1775	0.25	1419	0.80	588	0.41
Self-cons few-shot	5259	1283	0.24	1062	0.83	469	0.44
GenKnow 0-shot	4677	936	0.20	585	0.63	131	0.22
GenKnow 1-shot	4005	1012	0.25	752	0.74	326	0.43
GenKnow few-shot	4676	1271	0.27	1057	0.83	546	0.52
ReAct 0-shot	2550	428	0.20	246	0.57	78	0.32
ReAct 1-shot	5248	1321	0.26	932	0.71	415	0.45
ReAct few-shot	4674	1195	0.26	916	0.77	370	0.40

Table 8
Assessing novelty of the extraction of Mistral 7B

Prompt type	Triples linked to Wikidata		Triples already in Wikidata		Novel Triples	
	Nr	%	Nr	%	Nr	%
0-shot	443	0.19	64	0.14	379	0.86
1-shot	2590	0.51	478	0.18	2112	0.82
Few-shot	1953	0.49	360	0.18	1593	0.82
RAG 0-shot	1519	0.54	382	0.25	1137	0.75
RAG few-shot	1398	0.56	398	0.28	1000	0.72
CoT 0-shot	379	0.15	56	0.15	323	0.85
CoT 1-shot	1557	0.42	306	0.20	1251	0.80
CoT few-shot	1438	0.41	277	0.19	1161	0.81
Self-cons 0-shot	631	0.22	75	0.12	556	0.88
Self-cons 1-shot	2232	0.47	387	0.17	1845	0.83
Self-cons few-shot	2120	0.50	372	0.18	1748	0.82
GenKnow 0-shot	678	0.25	84	0.12	594	0.88
GenKnow 1-shot	1646	0.46	364	0.22	1282	0.78
GenKnow few-shot	1631	0.50	353	0.22	1278	0.78
ReAct 0-shot	345	0.21	39	0.11	306	0.89
ReAct 1-shot	1282	0.35	183	0.14	1099	0.86
ReAct few-shot	1227	0.40	254	0.21	973	0.79

Table 9
Assessing novelty of the extraction of Llama 3

Prompt type	Triples linked to Wikidata		Triples already in Wikidata		Novel Triples	
	Nr	%	Nr	%	Nr	%
0-shot	1046	0.26	74	0.07	972	0.93
1-shot	1744	0.44	245	0.14	1499	0.86
Few-shot	2380	0.47	441	0.19	1936	0.81
RAG 0-shot	3287	0.58	556	0.17	2731	0.83
RAG few-shot	3235	0.62	603	0.19	2632	0.81
CoT 0-shot	1005	0.11	73	0.07	932	0.93
CoT 1-shot	4658	0.54	824	0.18	3834	0.82
CoT few-shot	4365	0.58	714	0.16	3651	0.84
Self-cons 0-shot	1557	0.24	101	0.06	1456	0.94
Self-cons 1-shot	3602	0.51	564	0.16	3038	0.84
Self-cons few-shot	2703	0.51	317	0.12	2386	0.88
GenKnow 0-shot	1496	0.33	78	0.05	1418	0.95
GenKnow 1-shot	2062	0.52	327	0.16	1735	0.84
GenKnow few-shot	2633	0.56	519	0.20	2114	0.80
ReAct 0-shot	582	0.27	49	0.08	533	0.92
ReAct 1-shot	2235	0.43	373	0.17	1862	0.83
ReAct few-shot	1986	0.43	275	0.14	1711	0.86

Table 10
Analysis of extracted entities and relations by GPT-4

Prompt type	Entity Analysis			Relation Analysis				
	All	Linked		All	Linked		has Domain&Range	
	Nr	Nr	%	Nr	Nr	%	Nr	%
0-shot	4477	1672	0.37	1617	355	0.22	150	0.42
1-shot	4106	2539	0.62	1574	542	0.34	235	0.43
Few-shot	2996	2075	0.69	1156	469	0.41	207	0.44
RAG 1-shot	3712	2477	0.67	1497	604	0.40	268	0.44
RAG few-shot	3120	2348	0.75	1014	488	0.48	243	0.50
CoT 0-shot	4941	1571	0.32	1926	283	0.15	121	0.43
CoT 1-shot	3714	2602	0.70	1455	601	0.41	290	0.48
CoT few-shot	3057	2281	0.75	1111	562	0.51	265	0.47
Self-cons 0-shot	4301	1735	0.40	1614	278	0.17	121	0.44
Self-cons 1-shot	3661	2735	0.75	1320	524	0.40	245	0.47
Self-cons few-shot	3118	2509	0.80	984	506	0.51	241	0.48
GenKnow 0-shot	4421	1667	0.38	1661	305	0.18	127	0.42
GenKnow 1-shot	3579	2513	0.70	1525	547	0.36	266	0.49
GenKnow few-shot	3236	2440	.075	1131	514	0.45	241	0.47
ReAct 0-shot	4129	1629	0.39	1652	298	0.18	129	0.43
ReAct 1-shot	3284	2362	0.72	1314	536	0.41	255	0.48
ReAct few-shot	2716	2121	0.78	976	479	0.49	243	0.51

Table 11
Analysis of extracted entities and relations by Mistral 7B

Prompt type	Entity Analysis			Relation Analysis				
	All	Linked		All	Linked		has Domain&Range	
	Nr	Nr	%	Nr	Nr	%	Nr	%
0-shot	2745	1264	0.46	1289	294	0.23	127	0.43
1-shot	4174	2913	0.70	1532	659	0.43	277	0.42
Few-shot	3778	2712	0.72	1334	594	0.45	255	0.43
RAG 1-shot	3093	2365	0.76	1050	497	0.47	248	0.50
RAG few-shot	2750	2198	0.80	933	426	0.46	218	0.51
CoT 0-shot	3161	1276	0.40	1410	253	0.18	109	0.43
CoT 1-shot	3920	2667	0.68	1644	637	0.39	291	0.46
CoT few-shot	3781	2419	0.64	1638	658	0.40	303	0.46
Self-cons 0-shot	3188	1607	0.50	1525	325	0.21	127	0.39
Self-cons 1-shot	4277	2858	0.67	1851	757	0.41	356	0.47
Self-cons few-shot	4005	2887	0.72	1664	772	0.46	378	0.49
GenKnow 0-shot	3071	1637	0.53	1402	328	0.23	133	0.41
GenKnow 1-shot	3956	2593	0.66	1474	541	0.37	257	0.48
GenKnow few-shot	3620	2502	0.69	1172	476	0.41	215	0.45
ReAct 0-shot	2017	940	0.47	944	241	0.26	108	0.45
ReAct 1-shot	4158	2348	0.56	1911	663	0.35	307	0.46
ReAct few-shot	3351	2092	0.62	1505	566	0.38	268	0.47

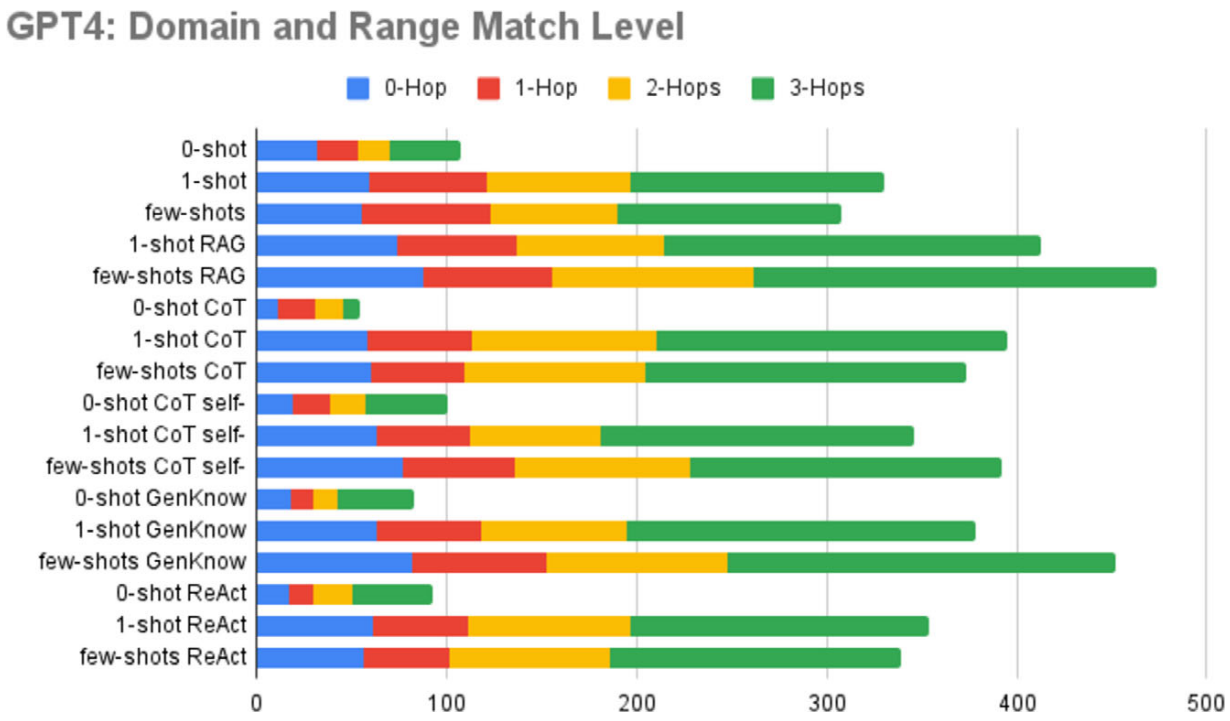

Fig. 5. Analysis of domain & range matching level of GPT-4.

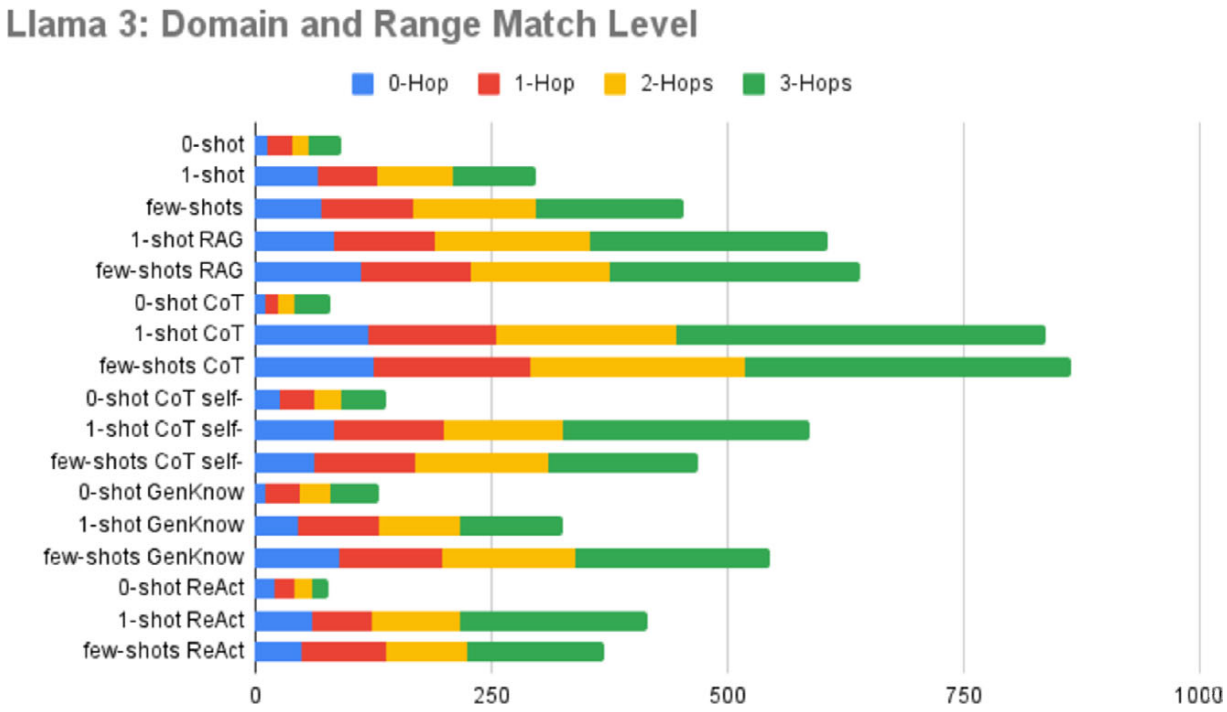

Fig. 6. Analysis of domain & range matching level of Llama 3.

Table 12
Precision, recall, F1 score

	Precision			Recall			F1 Score		
Prompt type	Mistral	Llama3	GPT-4	Mistral	Llama3	GPT-4	Mistral	Llama3	GPT-4
0-shot	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1-shot	0.02	0.02	0.01	0.08	0.06	0.04	0.04	0.03	0.02
Few-shot	0.03	0.02	0.02	0.08	0.06	0.06	0.04	0.03	0.03
RAG 1-shot	0.07	0.05	0.05	0.12	0.14	0.11	0.09	0.08	0.07
RAG few-shot	0.09	0.07	0.08	0.14	0.16	0.15	0.10	0.10	0.11
CoT 0-shot	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
CoT 1-shot	0.05	0.05	0.03	0.10	0.15	0.07	0.06	0.07	0.05
CoT few-shot	0.04	0.03	0.05	0.09	0.11	0.08	0.06	0.05	0.06
Self-cons 0-shot	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Self-cons 1-shot	0.05	0.05	0.03	0.10	0.13	0.06	0.06	0.07	0.04
Self-cons few-shot	0.04	0.02	0.05	0.13	0.08	0.10	0.07	0.03	0.06
GenKnow 0-shot	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
GenKnow 1-shot	0.05	0.03	0.03	0.12	0.09	0.07	0.07	0.05	0.04
GenKnow few-shot	0.05	0.04	0.07	0.12	0.11	0.13	0.07	0.06	0.09
ReAct 0-shot	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ReAct 1-shot	0.03	0.02	0.03	0.07	0.07	0.06	0.04	0.04	0.04
ReAct few-shot	0.04	0.02	0.06	0.09	0.08	0.09	0.06	0.04	0.07

References

[1] AI@Meta, Llama 3 Model Card, 2024, https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md.

[2] D. Ammalainen, 2023, https://upload.wikimedia.org/wikipedia/commons/1/1b/Wikidata_ontology_issues_---_suggestions_for_prioritisation_2023.pdf.

[3] T. Brown, B. Mann, N. Ryder, M. Subbiah, J.D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever and D. Amodei, Language models are few-shot learners, in: *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan and H. Lin, eds, Vol. 33, Curran Associates, Inc., 2020, pp. 1877–1901, https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[4] J. Carbonell and J. Goldstein, The use of MMR, diversity-based reranking for reordering documents and producing summaries, in: *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'98*, Association for Computing Machinery, New York, NY, USA, 1998, pp. 335–336. ISBN 1581130155. doi:10.1145/290941.291025.

[5] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H.W. Chung, C. Sutton, S. Gehrmann, P. Schuh, K. Shi, S. Tsvyashchenko, J. Maynez, A. Rao, P. Barnes, Y. Tay, N. Shazeer, V. Prabhakaran, E. Reif, N. Du, B. Hutchinson, R. Pope, J. Bradbury, J. Austin, M. Isard, G. Gur-Ari, P. Yin, T. Duke, A. Levskaya, S. Ghemawat, S. Dev, H. Michalewski, X. Garcia, V. Misra, K. Robinson, L. Fedus, D. Zhou, D. Ippolito, D. Luan, H. Lim, B. Zoph, A. Spiridonov, R. Sepassi, D. Dohan, S. Agrawal, M. Omernick, A.M. Dai, T.S. Pillai, M. Pellat, A. Lewkowycz, E. Moreira, R. Child, O. Polozov, K. Lee, Z. Zhou, X. Wang, B. Saeta, M. Diaz, O. Firat, M. Catasta, J. Wei, K. Meier-Hellstern, D. Eck, J. Dean, S. Petrov and N. Fiedel, *PaLM: Scaling Language Modeling with Pathways*, 2022.

[6] P.F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg and D. Amodei, Deep reinforcement learning from human preferences, in: *Advances in Neural Information Processing Systems*, I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan and R. Garnett, eds, Vol. 30, Curran Associates, Inc., 2017, https://proceedings.neurips.cc/paper_files/paper/2017/file/d5e2c0adad503c91f91df240d0cd4e49-Paper.pdf.

[7] S. Gehrmann, T. Adewumi, K. Aggarwal, P.S. Ammanamanchi, A. Aremu, A. Bosselut, K.R. Chandu, M.-A. Clinciu, D. Das, K. Dhole, W. Du, E. Durmus, O. Duš, C.C. Emezue, V. Gangal, C. Garbacea, T. Hashimoto, Y. Hou, Y. Jernite, H. Jhamtani, Y. Ji, S. Jolly, M. Kale, D. Kumar, F. Ladhak, A. Madaan, M. Maddela, K. Mahajan, S. Mahamood, B.P. Majumder, P.H. Martins, A. McMillan-Major, S. Mille, E. van Miltenburg, M. Nadeem, S. Narayan, V. Nikolaev, A. Niyongabo Rubungo, S. Osei, A. Parikh, L. Perez-Beltrachini, N.R. Rao, V. Raunak, J.D. Rodriguez, S. Santhanam, J. Sedoc, T. Sellam, S. Shaikh, A. Shimorina, M.A. Sobrevilla Cabezudo, H. Strobelt, N. Subramani, W. Xu, D. Yang, A. Yerukola and J. Zhou, The GEM benchmark: Natural language generation, its evaluation and metrics, in: *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, A. Bosselut, E. Durmus, V.P. Gangal, S. Gehrmann, Y. Jernite, L. Perez-Beltrachini, S. Shaikh and W. Xu, eds, Association for Computational Linguistics, Online, 2021, pp. 96–120. doi:10.18653/v1/2021.gem-1.10.

[8] A. Hogan, E. Blomqvist, M. Cochez, C. D'amato, G.D. Melo, C. Gutierrez, S. Kirrane, J.E.L. Gayo, R. Navigli, S. Neumaier, A.-C.N. Ngomo, A. Polleres, S.M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab and A. Zimmermann, *Knowledge Graphs, ACM Comput. Surv.* **54**(4) (2021). doi:10.1145/3447772.

[9] P.-L. Huguet Cabot and R. Navigli, REBEL: Relation extraction by end-to-end language generation, in: *Findings of the Association for Computational Linguistics: EMNLP 2021*, M.-F. Moens, X. Huang, L. Specia and S.W.-T. Yih, eds, Association for Computational Linguistics, Punta Cana, Dominican Republic, 2021, pp. 2370–2381. doi:10.18653/v1/2021.findings-emnlp.204.

[10] P.-L. Huguet Cabot, S. Tedeschi, A.-C.N. Ngomo and R. Navigli, RED$^{\text{FM}}$: A filtered and multilingual relation extraction dataset, in: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, A. Rogers, J. Boyd-Graber and N. Okazaki, eds, Association for Computational Linguistics, Toronto, Canada, 2023, pp. 4326–4343. doi:10.18653/v1/2023.acl-long.237.

[11] A.Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D.S. Chaplot, D.D.L. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier et al., Mistral 7B, 2023, arXiv preprint arXiv:2310.06825.

[12] M. Josifoski, N. De Cao, M. Peyrard, F. Petroni and R. West, GenIE: Generative information extraction, in: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, M. Carpuat, M.-C. de Marneffe and I.V. Meza Ruiz, eds, Association for Computational Linguistics, Seattle, United States, 2022, pp. 4626–4643. doi:10.18653/v1/2022.naacl-main.342.

[13] M. Josifoski, M. Sakota, M. Peyrard and R. West, Exploiting Asymmetry for Synthetic Training Data Generation: SynthIE and the Case of Information Extraction, 2023, arXiv preprint arXiv:2303.04132.

[14] M. Khalifa, L. Logeswaran, M. Lee, H. Lee and L. Wang, Exploring demonstration ensembling for in-context learning, in: *ICLR 2023 Workshop on Mathematical and Empirical Understanding of Foundation Models*, 2023, https://openreview.net/forum?id=9kK4R_8nAsD.

[15] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov and L. Zettlemoyer, BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, in: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter and J. Tetreault, eds, Association for Computational Linguistics, Online, 2020, pp. 7871–7880. doi:10.18653/v1/2020.acl-main.703.

[16] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Kü, M. Lewis, W.-T. Yih, T. Rocktä, S. Riedel and D. Kiela, Retrieval-augmented generation for knowledge-intensive NLP tasks, in: *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan and H. Lin, eds, Vol. 33, Curran Associates, Inc., 2020, pp. 9459–9474, https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf.

[17] B. Li, G. Fang, Y. Yang, Q. Wang, W. Ye, W. Zhao and S. Zhang, Evaluating ChatGPT's Information Extraction Capabilities: An Assessment of Performance, Explainability, Calibration, and Faithfulness, (2023), arXiv:2304.11633. https://api.semanticscholar.org/CorpusID:258297899

[18] J. Liu, A. Liu, X. Lu, S. Welleck, P. West, R. Le Bras, Y. Choi and H. Hajishirzi, Generated knowledge prompting for commonsense reasoning, in: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, S. Muresan, P. Nakov and A. Villavicencio, eds, Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 3154–3169. doi:10.18653/v1/2022.acl-long.225.

[19] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi and G. Neubig, Pre-Train, Prompt, and Predict: A Systematic Survey of Prompting Methods in, *Natural Language Processing, ACM Comput. Surv.* **55**(9) (2023). doi:10.1145/3560815.

[20] J.L. Martinez-Rodriguez, A. Hogan and I. Lopez-Arevalo, Information extraction meets the semantic web: A survey, *Semantic Web* **11**(2) (2020), 255–335. doi:10.3233/SW-180333.

[21] B. Min, H. Ross, E. Sulem, A.P.B. Veyseh, T.H. Nguyen, O. Sainz, E. Agirre, I. Heintz and D. Roth, *Recent Advances in Natural Language Processing via Large Pre-Trained Language Models: A Survey*, ACM Comput. Surv, 2023, Just Accepted. doi:10.1145/3605943.

[22] OpenAI, GPT-4 Technical Report, 2023.

[23] F. Petroni, T. Rocktäschel, S. Riedel, P. Lewis, A. Bakhtin, Y. Wu and A. Miller, Language models as knowledge bases? 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)*, K. Inui, J. Jiang, V. Ng and X. Wan, eds, Association for Computational Linguistics, Hong Kong, China, 2019, pp. 2463–2473. doi:10.18653/v1/D19-1250.

[24] E. Saravia, 2022, https://www.promptingguide.ai/.

[25] H. Su, W. Shi, J. Kasai, Y. Wang, Y. Hu, M. Ostendorf, W.-T. Yih, N.A. Smith, L. Zettlemoyer and T. Yu, One embedder, any task: Instruction-finetuned text embeddings, in: *Findings of the Association for Computational Linguistics: ACL 2023*, A. Rogers, J. Boyd-Graber and N. Okazaki, eds, Association for Computational Linguistics, Toronto, Canada, 2023, pp. 1102–1121. doi:10.18653/v1/2023.findings-acl.71.

[26] R. Thoppilan, D.D. Freitas, J. Hall, N. Shazeer, A. Kulshreshtha, H.-T. Cheng, A. Jin, T. Bos, L. Baker, Y. Du, Y. Li, H. Lee, H.S. Zheng, A. Ghafouri, M. Menegali, Y. Huang, M. Krikun, D. Lepikhin, J. Qin, D. Chen, Y. Xu, Z. Chen, A. Roberts, M. Bosma, V. Zhao, Y. Zhou, C.-C. Chang, I. Krivokon, W. Rusch, M. Pickett, P. Srinivasan, L. Man, K. Meier-Hellstern, M.R. Morris, T. Doshi, R.D. Santos, T. Duke, J. Soraker, B. Zevenbergen, V. Prabhakaran, M. Diaz, B. Hutchinson, K. Olson, A. Molina, E. Hoffman-John, J. Lee, L. Aroyo, R. Rajakumar, A. Butryna, M. Lamm, V. Kuzmina, J. Fenton, A. Cohen, R. Bernstein, R. Kurzweil, B. Aguera-Arcas, C. Cui, M. Croak, E. Chi and Q. Le, *LaMDA: Language Models for Dialog Applications*, 2022.

[27] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Roziè, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave and G. Lample, *LLaMA: Open and Efficient Foundation Language Models*, 2023.

[28] D. Vrandeč and M. Krötzsch, Wikidata: A free collaborative knowledgebase, *Communications of the ACM* **57**(10) (2014), 78–85. doi:10.1145/2629489.

[29] S. Wadhwa, S. Amir and B. Wallace, 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)*, A. Rogers, J. Boyd-Graber and N. Okazaki, eds, Association for Computational Linguistics, Toronto, Canada, 2023, pp. 15566–15589. doi:10.18653/v1/2023.acl-long.868.

[30] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery and D. Zhou, Self-consistency improves chain of thought reasoning in language models, in: *International Conference on Learning Representations (ICLR)*, 2023.

[31] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q.V. Le and D. Zhou, Chain-of-thought prompting elicits reasoning in large language models, in: *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho and A. Oh, eds, Vol. 35, Curran Associates, Inc., 2022, pp. 24824–24837, https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf.

[32] C. Whitehouse, C. Vania, A.F. Aji, C. Christodoulopoulos and A. Pierleoni, WebIE: Faithful and robust information extraction on the web, in: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, A. Rogers, J. Boyd-Graber and N. Okazaki, eds, Association for Computational Linguistics, Toronto, Canada, 2023, pp. 7734–7755. doi:10.18653/v1/2023.acl-long.428.

[33] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan and Y. Cao, ReAct: Synergizing reasoning and acting in language models, in: *International Conference on Learning Representations (ICLR)*, 2023.

[34] X. Ye, S. Iyer, A. Celikyilmaz, V. Stoyanov, G. Durrett and R. Pasunuru, Complementary explanations for effective in-context learning, in: *Findings of the Association for Computational Linguistics: ACL 2023*, A. Rogers, J. Boyd-Graber and N. Okazaki, eds, Association for Computational Linguistics, Toronto, Canada, 2023, pp. 4469–4484. doi:10.18653/v1/2023.findings-acl.273.

[35] D. Zeng, K. Liu, S. Lai, G. Zhou and J. Zhao, Relation classification via convolutional deep neural network, in: *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, J. Tsujii and J. Hajic, eds, Dublin City University and Association for Computational Linguistics, Dublin, Ireland, 2014, pp. 2335–2344, https://aclanthology.org/C14-1220.

[36] S. Zhang, L. Dong, X. Li, S. Zhang, X. Sun, S. Wang, J. Li, R. Hu, T. Zhang, F. Wu and G. Wang, Instruction Tuning for Large Language Models: A Survey, 2023.

[37] S. Zheng, Y. Hao, D. Lu, H. Bao, J. Xu, H. Hao and B. Xu, Joint entity and relation extraction based on a hybrid neural network, *Neurocomputing* **257** (2017), 59–66. doi:10.1016/j.neucom.2016.12.075.