Entity Quality Enhancement in Knowledge Graphs through LLM-based Question Answering

Morteza Kamaladdini Ezzabady[1], Farah Benamara[1,2]
[1]IRIT, University of Toulouse, France, [2]IPAL, CNRS-NUS-A*STAR, Singapore
{morteza.ezzabady,farah.benamara}@irit.fr

Abstract

Most models for triple extraction from texts primarily focus on named entities. However, real-world applications often comprise non-named entities that pose serious challenges for entity linking and disambiguation. We focus on these entities and propose the first LLM-based entity revision framework to improve the quality of extracted triples via a multi-choice question-answering mechanism. When evaluated on two benchmark datasets, our results show a significant improvement, thereby generating more reliable triples for knowledge graphs.

1 Introduction

Triple extraction (TE) is a well-established NLP task where several deep learning models (Bouziani et al., 2024; Wang et al., 2022; Santosh et al., 2021; Wadhwa et al., 2023; Xu et al., 2023), and more recently, LLMs (Trajanoska et al., 2023; Chia et al., 2022; Li et al., 2024; Chen et al., 2023) have successfully been employed in benchmark datasets in different domains and languages (e.g., SemEval-2010 Task 8 (Hendrickx et al., 2010), TACRED (Zhang et al., 2017), BioRed (Luo et al., 2022)).

Most relation extraction models focus primarily on named entities such as person names, locations, and organizations, making them fail in dealing with a richer array of complex, non-named entities (hearafter N-NE). According to Paris and Suchanek (2021), N-NE are defined as noun phrases (NPs) that can be the subject or object of a predicate within a sentence such as "decision list" and "parsing-based ne rules" in Figure 1. N-NE can have several forms ranging from nominal group (e.g., *year 1944*), containing adjectives and adverbs (e.g., *very good questions*), prepositional phrases (e.g., *in the Arab World*), relative clauses or more complex syntactic constructions (e.g., *near-term growth prospects of the global economy*). N-NE are relatively frequent in textual data. For example, when manually analysing around 2K NPs extracted from Wikipedia, Paris and Suchanek (2021) found that 78% of NP heads are N-NE among which 38% are modified by an adjective, and 34% have a preposition. Despite their importance, their frequency in popular benchmark datasets is relatively low (e.g., TACRED only involves named entities).

Context: *first, decision list is used to learn the parsing-based ne rules.*

Gold: ('decision list', 'usage', 'parsing-based ne rules')
GPT-4: ('decision list', 'usage', 'learn the parsing-based ne rules')
Falcon-2: ('first, decision list', 'learn', 'parsing-based ne rules')

Figure 1: Triple extraction involving N-NE as given by gold manual annotations, Falcon-2 and GPT4 models. Wrong entities are in red.

N-NE pose serious challenges in knowledge graph (KG) construction and reasoning, because they remain *silent* with no chance to be linked into an existing knowledge bases (KBs) such as YAGO4 (Tanon et al., 2020) or Wikidata (Vrandečić, 2012). Figure 1 illustrates the impact these entities have on triple extraction from a sentence taken for SemEval 2018 Task 7. We compare the outputs of Falcon-2 (Sakor et al., 2020), a entity and relation linking tool over Wikidata, and zero-shot GPT-4 against the gold label. Although both models successfully identified the boundaries of the entities, they failed to correctly extract both the head and tail entities together.

N-NE have received little attention in the literature. Among the few works, Open Information Extraction tools such as OpenIE (Angeli et al., 2015) (see (Zhou et al., 2022) for a survey) output triples of subject, predicate, and object in an unsupervised way relaying on dependency parsers where relation arguments can contain N-NE. Paris and Suchanek (2021) performed a qualitative manual study of the nature of N-NE in Wikipedia. In this paper, we go one step further by proposing, for the first time as far as we know, **an end to end LLM-based entity**

Proceedings of the Generative AI and Knowledge Graph Workshop (GenAIK), pages 136–145
January 19, 2025. ©2025 International Committee on Computational Linguistics (ICCL)

revision framework that (a) automatically extracts triples from raw texts, (b) identifies N-NE, (c) enhances their quality by augmenting their likelihood of being successfully linked to an external knowledge base, which is a first important step to overall KG quality assessment (Chen et al., 2019).

To this end, we adopt a multiple choice prompting (MCP) strategy on top of a triple extractor to verify the extracted entities. MCP has been successfully used as a self-evaluation method to mitigate LLMs errors in complex problems like arithmetic and commonsense reasoning (Miao et al., 2023; Weng et al., 2023; Ren et al., 2023). It is newly employed here for entity quality enhancement. Our contributions are as follows:

1. A multiple-choice question answering (MCQA) strategy for enhancing LLMs to revise their extracted entities,

2. Comprehensive experiments with both open source and closed LLMs on two benchmark datasets for relation extraction,

3. A manual analysis of our results demonstrating the effectiveness of our framework in correctly identifying and selecting N-NE.

This paper is organized as follows. Section 2 presents our overall framework, Section 3 details the datasets used for evaluation, the experimental settings and evaluation metrics. We finally gives our results together with an error analysis in Section 4.

2 Entity Revision through LLM-based Question Answering

Figure 3 shows our three-steps framework: (1) It first extracts triples using an in-context learning approach. (2) It then ranks candidate entities and (3) refines entity selection through a multiple-choice format to improve accuracy by learning from common extraction errors.

It is important to note that our framework has been designed with modularity in mind, independently from the method used for triple extraction and how N-NE are initially identified. However as a first step and in order to evaluate the effectiveness of our approach when evaluated on benchmark datasets, we experiment with target relations as input to Step 1, the subsequent steps are agnostic to this guidance. This allows to increase the number of matching triples generated by LLMs when compared to gold annotations (see below) and therefore ensure a sufficient number of instances to derive meaningful conclusions (see Section 3.3 about the evaluation protocols). We detail below each step.

2.1 Step 1: Triple Extraction and Matching

We instruct the LLMs to extract triples via an in-context learning method following (Ozyurt et al., 2024; Lyu et al., 2023; Ma et al., 2023a) where prompts only contain the definition of the target relation. Given is a set of contexts $\mathcal{C} = \{c_i\}$. For each context c_i, the aim is to enumerate triples $\{(h_{ij}, r_{ij}, t_{ij})\}_{j=1}^{R_i}$, where $r_{ij} \in \mathcal{R}$ is a relation and h_{ij} and t_{ij} are the head and tail entities for the relation r_{ij}, and where R_i is the number of relations in c_i (cf. Figure 2).

Step 1 is evaluated by matching LLMs generated triples to gold ones based on overlapping entities. For instance, the gold triple for the given context in Figure 2 is `(global variables, USED-FOR, global properties)`, of which only the extracted triple `(global variables, USED-FOR, representing global properties)` matches the gold standard.

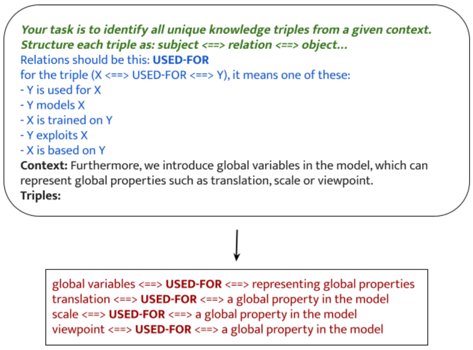

Figure 2: Example of prompt used for triple extraction. The green, blue and black in the top box represent the instruction, demonstration and test context in the prompt respectively. The red parts are the LLMs outputs.

2.2 Step 2: Candidates Selection

Let $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$ be a knowledge graph, where $\mathcal{E}$ is the set of entities, $\mathcal{R}$ the set of relations, and $\mathcal{T} = \{(h, r, t) | h \in \mathcal{E}, r \in \mathcal{R}, t \in \mathcal{E}\}$ the set of triples. Given a query $(h, r, ?)$ (resp. $(?, r, t)$), the graph completion task ranks each entity by calculating its score to determine how well it makes the query hold, thereby achieving knowledge graph completion (Wei et al., 2023). This task inspired our approach; however, as we do not possess a predefined set of entities, we must generate a list of

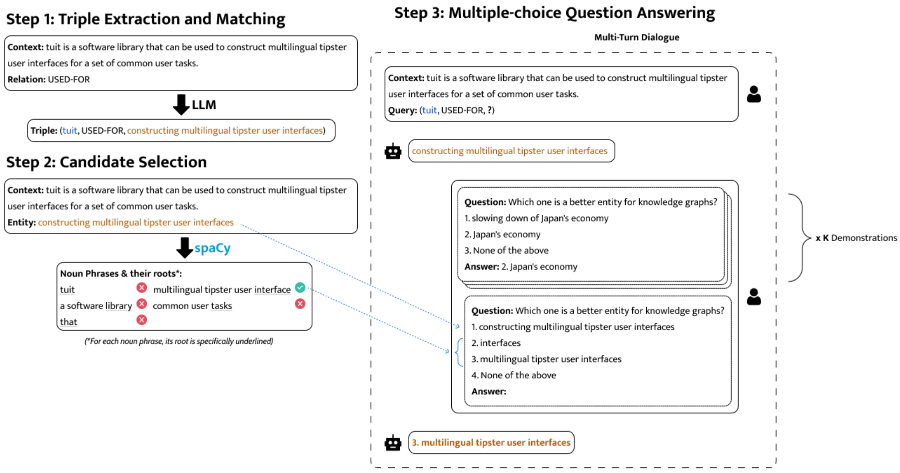

Figure 3: Overview of our entity revision framework: (1) **Triple extraction** from the given context to identify relevant relationships; (2) **Candidate selection**, where potential entities are shortlisted as relevant targets; (3) **Multiple-choice question-answering** to determine the most suitable entity.

potential candidates to fill the queries $(h_{ij}, r_{ij}, ?)$ (resp. $(?, r_{ij}, t_{ij})$) based on context c_i and utilize LLMs as a ranker.

Our candidate selector relies on SpaCy [1] parser,[2] known for its fast and accurate syntactic analysis, to select all noun phrases from context c_i that either contain the entity t_{ij} (resp. h_{ij}) or are contained by t_{ij} (resp. h_{ij}), along with the root of those noun phrases. This method ensures that the selected candidates are contextually relevant and are more likely to be correct entities that can replace low-quality extracted entities. Step 2 is then evaluated by checking if the selected candidate entities include the gold entities or not.

2.3 Step 3: Multiple Choice Question Answering (MCQA).

LLMs are generally not effective as few-shot information extractors, but they excel as rankers Ma et al. (2023b). We therefore employ prompting strategies similar to QA4RE (Zhang et al., 2023), transforming our task into multi-choice questions to more accurately select entities.

To enhance entity extraction, we utilize a set of K demonstration examples that target common extraction errors. These include entities mistakenly containing verbs, excessive adjectives, pronouns, determiners, and pseudo-sentences. Such errors often lead to inaccuracies in the model's outputs, particularly in sentences where the distance between head and tail entities in the context is long (Xu et al., 2023; Ezzabady et al., 2024). Following Mo et al. (2024) that use direct comparisons to better guide LLMs, each example is selected based on its ability to clearly demonstrate these specific issues, offering a dual presentation of both incorrect and correct entity identifications.

Here are our demonstration questions-answer pairs.

Verb phrase
```
Question: Which one is a better entity
for knowledge graphs?
1. slowing down of Japan's economy
2. Japan's economy
3. None of the above
Answer: 2. Japan's economy
```

Redundant adjective
```
Question: Which one is a better entity
in a knowledge graph?
1. sars-cov-2 outbreak
2. outbreak
3. large sars-cov-2 outbreak
4. None of the above
Answer: 1. sars-cov-2 outbreak
```

Determiner
```
Question: Which one is a better entity
in a knowledge graph?
1. identification
2. both language identification
3. language identification
```

[1] https://spacy.io/

[2] Although this step could also be performed by LLMs, we opted to use SpaCy here to keep the LLM more focused on the entity revision task.

```
4. None of the above
Answer: 3. language identification
```

Pronoun

```
Question: Which one is a better entity
in a knowledge graph?
1. application
2. My application
3. None of the above
Answer: 1. application
```

None of the above

```
Question: Which one is a better entity in
a knowledge graph?
1. keep inflation high in the near term
2. keep inflation high
3. None of the above
Answer: 3. None of the above
```

3 Experiments

3.1 Datasets

As far as we know, only two benchmark relation extraction datasets involving N-NE exist: SemEval 2018 Task 7 (Gábor et al., 2018) and SciERC (Luan et al., 2018). Both are *document-based* datasets annotated for entities and their relations extracted from scientific abstracts. They are a good choice to evaluate our framework (see Table 1) as their triples contain less than 5% of named entities (as given by SpaCy) and more importantly less than 35% are linked to Wikidata. This is also aligned with recent work by Zhu et al. (2024) who showed that SciERC is a challenge for making knowledge graphs, so that the performance of the best model (GPT-4) is less than 10%.

Gold Triples	SemEval 1,595			SciERC 4,265		
	Head	Tail	Both	Head	Tail	Both
% Named entities	3.71	2.13	0.13	4.71	3.42	0.49
% Linked with Wikidata	35.05	31.97	13.29	29.00	28.07	8.30

Table 1: SemEval and SciERC datasets statistics.

3.2 Experimental Settings

To increase triple matching and simplify the process for LLMs, we narrow down each document to sentences such that our input is a set of sentences $\{s|s \in d, h \in s, t \in s\}$.[3] This leads to a total of 1,578 sentences for SemEval and 4,151 for SciERC. For the inter-sentence relations (1.07% and 2.67% of triples in SemEval and SciERC respectively), we employ their documents as context.

Position bias and *No answer is true* are well known issues in MCQA with language models (Robinson et al., 2023). To address them, we follow the solutions proposed by Ren et al. (2023) as follows. We employ **shuffle and average** method that de-bias and correct answer position effects. To handle cases where none of the provided answers may be correct, we introduce a **None of the Above** option into the answer set, enhancing the model's ability to avoid overconfident incorrect predictions.

For our experiments, we rely on GPT-4,[4] LLaMA-3.1 8B-instruct[5] and Mistral 7B-instruct.[6] We compare our MCQA framework against two baselines:[7]

(a) **LLM with simple prompt (*simple*)**: which is similar to zero-shot learning where only the description of the task is given,

(b) **LLM with detailed prompt (*detailed*)**: that provides in addition a definition of what are considered to be good entities for a KG.

To demonstrate the superiority of our method over having specific guideline, we applied our method only on the *simple* baseline (hearafter ***simple+MCQA***). Both baselines operate in a zero-shot setting, MCQA being a few-shot prompting strategy where demonstration question-answer pairs are used to instruct the LLMs.

In Figure 4, we provide examples for different prompts as input and the corresponding output from GPT-4. In dialogues with LLMs, there are three key roles: the **System** role, which sets how the model answers; the **User** role, representing the individual who inputs queries; and the **Assistant** role, which encompasses the model's responses to user inputs. These roles collectively ensure a structured and effective interaction. A multi-turn dialogue involves a series of exchanges between the user and the assistant where each response builds on the previous interaction.

For all the models, and to avoid bias the same prompts have been used and more importantly,

[3] We also tested using documents as input, but the outcomes were inconclusive, e.g., in SciERC, the match rates for documents vs. sentences were 33.95% vs. 54.14%, respectively.

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

[5] https://llama.meta.com/llama3

[6] https://ollama.com/library/mistral:7b

[7] As this work focuses on improving LLMs performance, non-LLM methods are out of the scope of this paper.

demonstration questions were not sourced from evaluation datasets (cf. Section 2.3). Additionally, we set the number of demonstrations K to 4.[8] For implementation details see Appendix A.

3.3 Evaluation Protocol

We evaluate the performances in terms of four metrics, each metric aims to evaluate a particular step of our approach:

(a) Matched Triples. It counts the number of extracted triples from Step 1 that successfully match with at least one corresponding gold-standard triple. The matching is determined based on an overlap function, where a partial or complete overlap between the extracted and gold triples is sufficient to consider them matched. This metric provides an initial measure of how accurately the system can identify potential relationships from the context. For example, in Figure 4, none of the extracted triples via *detailed* prompt matches with the gold triple `(words, PART_WHOLE, corpus)`.

(b) Candidate selector success rate. It evaluates the effectiveness of the candidate selection step (Step 2). Specifically, it measures how often the true gold-standard entity is included among the set of candidates presented during the selection process. A candidate selection is successful when the gold entity is present in the generated options. This metric highlights the robustness of the candidate generation process and its ability to retain contextually relevant entities for further refinement. For example in Figure 4, we can observe that the candidate selector in our *simple+MCQA* method successfully included the gold entity `"words"` as options for the question corresponding to the triple `(words, PART_WHOLE, corpus)`.

(c) Correct entities. This metric evaluates Step 3 and focuses on the quality of entities within matched triples. It counts the number of entities within these triples that exactly match the corresponding entities in the gold-standard triples. We consider matches of entities at the head, tail and both head and tail positions. This metric is essential for assessing how accurately the framework identifies both the head and tail entities in relation to their expected true values, providing insight into the precision of the extraction process. For the gold triple `(words, PART_WHOLE, corpus)` from Figure 4, the outputs of the *simple* baseline and our *simple+MCQA* approach are `100,000 words` and `words`, respectively, as head entities, with the latter being the correct entity.

(d) Linking coverage. This metric is used to evaluate the overall LLM-based revision framework. It computes the percentage of entities that are linked to Wikidata, the largest collaborative general knowledge graph with more than 52 million instances (Heist et al., 2020). For example, in the gold triple `(words, PART_WHOLE, corpus)` from Figure 4, the tail entity `corpus` was linked to the entity with ID `Q461183` in Wikidata. To this end, we rely on SpaCy entity linker module[9]

4 Results and Discussions

4.1 Overall Results

Results are shown in Table 2. GPT-4 demonstrates notable improvements post-revision across all metrics on both datasets, most significantly in the whole triple category (i.e., head, tail and both), where the performance scores in terms of correct entities, rise 11% for SemEval and 9% for SciERC.

Conversely, LLaMA-3 exhibits a general decline in performance after revision across all categories. An interesting observation holds for the detailed baseline where LLaMA-3 seems to handle guidelines better than GPT-4 in the SemEval dataset where the matched triples was 323 vs. 218 for GPT-4. This could suggest that despite its smaller size and simpler architecture, which might hinder the integration of sophisticated entity revision techniques, LLaMA-3 is more compliant with structured guidelines.

Mistral initially performs worse than both GPT-4 and LLaMA-3; however, by applying our revision framework, its results notably improve. For instance, we observe an increase in correct entities in the head, tail and both for both datasets (except the head in SciERC). More importantly, the linking coverage also increases in particular for entities in tail positions in the extracted triples.

Finally, our results show the variability in performances between different LLMs in the triple extraction step where GPT4 is the best achieving a matching triples of 87% and 54.13% in SemEval and SciERC, respectively. This finding is inline with recent studies in generative relation extrac-

[8]We tested several values of K$\in [1,\ 4]$ and 4 was the best.

[9]https://github.com/egerber/spaCy-entity-linker

Input:
System: You are an expert in Knowledge graphs. Your task is to identify all unique knowledge triples from a given context. Structure each triple as: subject <==> relation <==> object. Separate each triple with a new line. Ensure that entities(subjects and objects) are clear, distinct, and specific concepts or items.
Relations should be this: PART_WHOLE
for the triple (X <==> PART_WHOLE <==> Y), it means one of these:
- X is a part, a component of Y
- X is found in Y
- Y is built from/composed of X
User: Context: after several experiments, and trained with a little corpus of 100,000 words, the system guesses correctly not placing commas with a precision of 96% and a recall of 98%.
Triples:
Output:
Assistant:
experiments <==> PART_WHOLE <==> trained with a little corpus of 100,000 words
100,000 words <==> PART_WHOLE <==> trained with a little corpus of 100,000 words
system <==> PART_WHOLE <==> experiments
system <==> PART_WHOLE <==> trained with a little corpus of 100,000 words
placing commas <==> PART_WHOLE <==> system guesses
precision of 96% <==> PART_WHOLE <==> system guesses
recall of 98% <==> PART_WHOLE <==> system guesses

Triple Extraction with simple prompt

Input:
System: You are an expert in Knowledge graphs. Your task is to identify and construct knowledge triples from the given context.
Each triple should consist of a subject, a relation, and an object, formatted as (subject <==> relation <==> object).
Adhere to these specific guidelines:
Entities (Subjects and Objects): Select key nouns or noun phrases that are concrete, distinct, and specific. These entities should represent tangible concepts or items and must not include actions or verbs. Ensure the entities are clear and unambiguous.
Separation of Components: Each part of the triple (subject, relation, object) must be distinctly identifiable and separate from the others. Ensure clarity in the distinction between these components.
Avoiding Ambiguity: Ensure that the phrasing of the triples avoids ambiguity. The entities and relation should be straightforward, leaving no room for multiple interpretations.
Relation: PART_WHOLE
for the triple (X <==> PART_WHOLE <==> Y), it means one of these:
- X is a part, a component of Y
- X is found in Y
- Y is built from/composed of X
User: Context: after several experiments, and trained with a little corpus of 100,000 words, the system guesses correctly not placing commas with a precision of 96% and a recall of 98%.
Triples:
Output:
Assistant:
1. (experiments <==> PART_WHOLE <==> several experiments)
2. (corpus of 100,000 words <==> PART_WHOLE <==> system training)
3. (precision of 96% <==> PART_WHOLE <==> system performance)
4. (recall of 98% <==> PART_WHOLE <==> system performance)

Triple Extraction with detailed prompt

Input:
System: You are an expert in Knowledge graphs. Your task is to complete the given triple by outputting a list of candidate for the missing entity (represented by "?") according to the context.
Candidates should be found in the context.
enumerate the candidates.
User: Context: after several experiments, and trained with a little corpus of 100,000 words, the system guesses correctly not placing commas with a precision of 96% and a recall of 98%.
Triple: (? <==> PART_WHOLE <==> trained with a little corpus of 100,000 words)
Assistant:
100,000 words
User: Follow the example:
Question: Which one is a better entity in a knowledge graph?
1. identification
2. both language identification
3. language identification
4. None of the above
Answer: 3. language identification
Question: Which one is a better entity in a knowledge graph?
1. words
2. 100,000 words
3. None of the above
Answer:
Output:
Assistant:
1. words

Multiple Choice Question Answering

Figure 4: Prompts and responses for our three models: LLM with simple prompt (the first baseline on the left), LLM with detailed prompt (second baseline in the middle) and our framework (on the right, for the sake of readability we only put one demonstration).

Method	**SemEval** (1,595 gold triples)				**SciERC** (4,265 gold triples)			
	Matched Triples	Correct Head	Correct Tail	Correct Both	Matched Triples	Correct Head	Correct Tail	Correct Both
LLaMA-3 (simple)	310	**232** (**8.40**)	210 (5.39)	**171** (**2.95**)	957	**688** (**8.75**)	**551** (4.60)	**435** (1.83)
LLaMA-3 (detailed)	**323**	223 (6.46)	**211** (4.76)	161 (2.13)	**1,010**	545 (8.02)	472 (4.95)	283 (**1.95**)
LLaMA-3 (simple + MCQA)	310 [75]	215 (7.08)	199 (**5.77**)	137 (2.38)	957 [70]	621 (6.80)	542 (**5.89**)	381 (1.74)
Mistral (simple)	**191**	124 (**2.19**)	92 (1.38)	68 (0.25)	**677**	**461** (**2.58**)	307 (1.74)	229 (**0.47**)
Mistral (detailed)	106	91 (1.76)	79 (1.07)	69 (**0.38**)	263	211 (1.34)	190 (0.94)	153 (0.19)
Mistral (simple + MCQA)	**191 [72]**	**128** (1.88)	**120** (**1.82**)	**82** (0.31)	**677 [70]**	441 (2.30)	**354** (**1.85**)	**245** (0.38)
GPT-4 (simple)	**1,384**	694 (12.92)	833 (13.98)	454 (1.15)	**2,309**	1,745 (10.34)	1,137 (6.61)	935 (1.95)
GPT-4 (detailed)	218	159 (2.63)	93 (1.00)	79 (0.25)	1,948	1,106 (9.87)	850 (6.54)	547 (1.85)
GPT-4 (simple + MCQA)	**1,384 [79]**	**850** (**19.94**)	**890** (**18.37**)	**609** (**5.39**)	**2,309 [79]**	**1,794** (**10.88**)	**1,408** (**10.39**)	**1,142** (**2.30**)

Table 2: Overall results of our LLM-based revision framework, in terms of: (a) Matched triples and Correct entities in the head, tail and both: number of instances, (b) Linking coverage: percentages between (), (c) Candidate selector success rates: percentages between []. The best scores per LLM are in bold font whereas best overall results are underlined. Please note that candidate selector success only concerns simple+MCQA as the baselines do not perform any selection.

tion (see for example (Jiang et al., 2024)). The second variability concerns LLMs performances when applying the MCQA technique. While the method demonstrates strong results with models like GPT-4 due to its advanced contextual reasoning and comprehension capabilities, it does not show similar improvements with models such as LLaMA-3. This inconsistency points to potential limitations in model architecture and pre-training data, which may affect how effectively they handle MCQA tasks. Future work should investigate these disparities to understand the specific features that

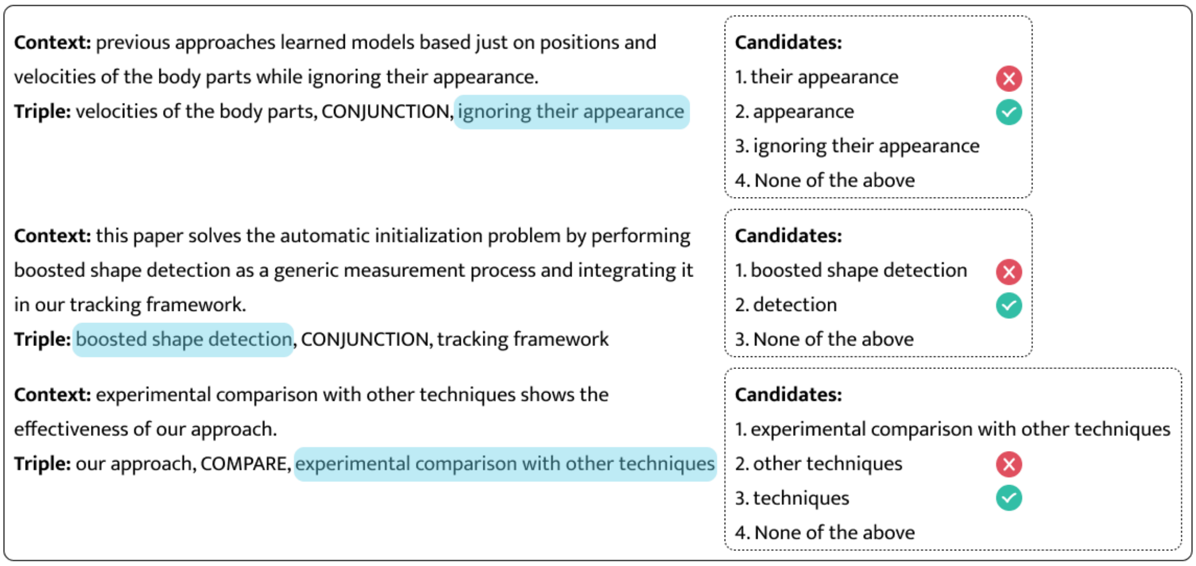

Figure 5: An example of errors made by our MCQA revision framework taken from the SciERC dataset. On the left, the output from triple extraction and matching (step 1) is displayed, highlighting the target entity that needs enhancement. On the right, we show a list of candidate entities obtained in step 2 (i.e., candidate selection) intended to refine the target entity. The entities predicted by our MCQA method answers (step 3) are incorrect (red cross), failing therefore in extracting the correct ones (marked by a green check).

enable some models to leverage MCQA successfully while identifying modifications or alternatives needed to improve performance in others.

4.2 Error Analysis

A manual error analysis of GPT-4 *simple+MPQA* outputs shows that our candidate selector missed 21% of the gold entities across both datasets. For the remaining 79%, the question-answering component achieved accuracies of 87% for SciERC and 81% for SemEval. Figure 5 shows some incorrect answers produced by our approach. Although providing demonstrations helped LLMs make better choices, the error categories (containing verbs, excessive adjectives, pronouns, determiners, and pseudo-sentences) have not been completely eliminated. For example, in the SciERC dataset using GPT-4, the number of entities containing verbs reduced from 737 to 352. Additionally, we observed that when LLMs are given inputs targeting multiple error categories (first example in Figure 5), they struggle to avoid all of them.

5 Conclusion

In this paper, we explore the potential of LLMs in-context learning for entity revision. To address the challenges posed by non-named entities, we introduced a multiple-choice question-answering framework that revises extracted entities from LLMs while increasing their linking coverage with the largest open knowledge base. When evaluated on two benchmark relation extraction datasets, our results demonstrate the effectiveness of our framework. We believe our work is a first important step to account for non-named entities in knowledge graph construction.

In this work, we apply a limited set of prompting techniques (zero-shot and few-shot in-context learning), which can be further explored in future research. We will also consider how improved entities affect downstream applications like question answering over knowledge graphs.

Acknowledgments

This work is jointly funded by the French National Agency QualityOnt ANR-21-CE23-0036-01, and the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 490998901.

Limitations

Our methodology is designed to be domain-agnostic, though it has initially been tested using two benchmark relation extraction datasets from the scientific domain in English. These datasets, selected due to the scarcity of benchmarks rich in non-named entities (N-NEs), offer a rigorous testbed for evaluating our approach's efficacy on N-NEs.

Despite their focus on scientific abstracts, our approach demonstrates the potential for broader applicability. Future research will expand this evaluation to include diverse datasets from various domains and languages, thereby providing a comprehensive assessment of the generalizability and robustness of our framework.

Our evaluation metric is based on the percentage of entities linked to Wikidata. Although it is the largest open knowledge graph in terms of the number of instances, some entities correctly retrieved by our model may be missed by the linking coverage metric simply because those entities do not exist in Wikidata. It will therefore be interesting to also measure the linking rate with other knowledge bases such as DBPedia and YAGO.

Ethics Statement

The data used for conducting the experiments are composed of scientific abstracts taken from datasets publicly available to the research community. The datasets do not contain offensive or abusive language. We utilized various large language models (LLMs), both open-source and proprietary. It is important to acknowledge that these LLMs can exhibit biases and may encounter challenges concerning factual accuracy. Therefore, a critical approach should be adopted when interpreting the experimental outcomes.

References

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 344–354, Beijing, China. Association for Computational Linguistics.

Nacime Bouziani, Shubhi Tyagi, Joseph Fisher, Jens Lehmann, and Andrea Pierleoni. 2024. REXEL: An End-to-end Model for Document-Level Relation Extraction and Entity Linking. *arXiv preprint*. ArXiv:2404.12788 [cs].

Haihua Chen, Gaohui Cao, Jiangping Chen, and Junhua Ding. 2019. A Practical Framework for Evaluating the Quality of Knowledge Graph. In *Knowledge Graph and Semantic Computing: Knowledge Computing and Language Understanding*, pages 111–122, Singapore. Springer.

Jiao Chen, Luyi Ma, Xiaohan Li, Nikhil Thakurdesai, Jianpeng Xu, Jason H. D. Cho, Kaushiki Nag, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. 2023. Knowledge Graph Completion Models are Few-shot Learners: An Empirical Study of Relation Labeling in E-commerce with LLMs. *arXiv preprint*. ArXiv:2305.09858 [cs].

Yew Ken Chia, Lidong Bing, Soujanya Poria, and Luo Si. 2022. RelationPrompt: Leveraging Prompts to Generate Synthetic Data for Zero-Shot Relation Triplet Extraction. In *Findings of the Association for Computational Linguistics: ACL 2022*.

Morteza Ezzabady, Frédéric Ieng, Hanieh Khorashadizadeh, Farah Benamara, Sven Groppe, and Soror Sahri. 2024. Towards generating high-quality knowledge graphs by leveraging large language models. In *The 29th Annual International Conference on Natural Language & Information Systems (NLDB 2024), Turin, Italy*.

Kata Gábor, Davide Buscaldi, Anne-Kathrin Schumann, Behrang QasemiZadeh, Haïfa Zargayouna, and Thierry Charnois. 2018. SemEval-2018 Task 7: Semantic Relation Extraction and Classification in Scientific Papers. In *Proceedings of the 12th International Workshop on Semantic Evaluation*, pages 679–688, New Orleans, Louisiana. Association for Computational Linguistics.

Nicolas Heist, Sven Hertling, Daniel Ringler, and Heiko Paulheim. 2020. Knowledge graphs on the web – an overview. *Preprint*, arXiv:2003.00719.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations between Pairs of Nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 33–38, Uppsala, Sweden. Association for Computational Linguistics.

Pengcheng Jiang, Jiacheng Lin, Zifeng Wang, Jimeng Sun, and Jiawei Han. 2024. GenRES: Rethinking evaluation for generative relation extraction in the era of large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2820–2837, Mexico City, Mexico. Association for Computational Linguistics.

Dawei Li, Zhen Tan, Tianlong Chen, and Huan Liu. 2024. Contextualization Distillation from Large Language Model for Knowledge Graph Completion. *arXiv preprint*. ArXiv:2402.01729 [cs].

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-Task Identification of Entities, Relations, and Coreference for Scientific Knowledge Graph Construction. *arXiv preprint*. ArXiv:1808.09602 [cs].

Ling Luo, Po-Ting Lai, Chih-Hsuan Wei, Cecilia N Arighi, and Zhiyong Lu. 2022. BioRED: a rich biomedical relation extraction dataset. *Briefings in Bioinformatics*, 23(5):bbac282.

Xinxi Lyu, Sewon Min, Iz Beltagy, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Z-ICL: Zero-Shot In-Context Learning with Pseudo-Demonstrations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2304–2317, Toronto, Canada. Association for Computational Linguistics.

Xilai Ma, Jing Li, and Min Zhang. 2023a. Chain of Thought with Explicit Evidence Reasoning for Few-shot Relation Extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2334–2352, Singapore. Association for Computational Linguistics.

Yubo Ma, Yixin Cao, YongChing Hong, and Aixin Sun. 2023b. Large Language Model Is Not a Good Few-shot Information Extractor, but a Good Reranker for Hard Samples! In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10572–10601. ArXiv:2303.08559 [cs].

Ning Miao, Yee Whye Teh, and Tom Rainforth. 2023. SelfCheck: Using LLMs to Zero-Shot Check Their Own Step-by-Step Reasoning. *arXiv preprint*. ArXiv:2308.00436 [cs].

Ying Mo, Jian Yang, Jiahao Liu, Shun Zhang, Jingang Wang, and Zhoujun Li. 2024. C-ICL: Contrastive In-context Learning for Information Extraction. *arXiv preprint*. ArXiv:2402.11254 [cs].

Yilmazcan Ozyurt, Stefan Feuerriegel, and Ce Zhang. 2024. Document-Level In-Context Few-Shot Relation Extraction via Pre-Trained Language Models. *arXiv preprint*. ArXiv:2310.11085 [cs] version: 2.

Pierre-Henri Paris and Fabian Suchanek. 2021. Non-named Entities – The Silent Majority. In Ruben Verborgh, Anastasia Dimou, Aidan Hogan, Claudia d'Amato, Ilaria Tiddi, Arne Bröring, Simon Mayer, Femke Ongenae, Riccardo Tommasini, and Mehwish Alam, editors, *The Semantic Web: ESWC 2021 Satellite Events*, volume 12739. Springer International Publishing, Cham. Series Title: Lecture Notes in Computer Science.

Jie Ren, Yao Zhao, Tu Vu, Peter J. Liu, and Balaji Lakshminarayanan. 2023. Self-Evaluation Improves Selective Generation in Large Language Models. *arXiv preprint*. ArXiv:2312.09300 [cs].

Joshua Robinson, Christopher Michael Rytting, and David Wingate. 2023. Leveraging Large Language Models for Multiple Choice Question Answering. *arXiv preprint*. ArXiv:2210.12353 [cs].

Ahmad Sakor, Kuldeep Singh, Anery Patel, and Maria-Esther Vidal. 2020. Falcon 2.0: An entity and relation linking tool over wikidata. In *Proceedings of the 29th ACM International Conference on Information Knowledge Management*, CIKM '20, page 3141–3148, New York, NY, USA. Association for Computing Machinery.

Tokala Yaswanth Sri Sai Santosh, Prantika Chakraborty, Sudakshina Dutta, Debarshi Kumar Sanyal, and Partha Pratim Das. 2021. Joint Entity and Relation Extraction from Scientific Documents: Role of Linguistic Information and Entity Types.

Thomas Pellissier Tanon, Gerhard Weikum, and Fabian Suchanek. 2020. YAGO 4: A Reason-able Knowledge Base. page 583.

Milena Trajanoska, Riste Stojanov, and Dimitar Trajanov. 2023. Enhancing Knowledge Graph Construction Using Large Language Models. *arXiv preprint*. ArXiv:2305.04676 [cs].

Denny Vrandečić. 2012. Wikidata: a new platform for collaborative data collection. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12 Companion, pages 1063–1064, New York, NY, USA. Association for Computing Machinery.

Somin Wadhwa, Silvio Amir, and Byron Wallace. 2023. Revisiting Relation Extraction in the era of Large Language Models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Chenguang Wang, Xiao Liu, Zui Chen, Haoyun Hong, Jie Tang, and Dawn Song. 2022. DeepStruct: Pretraining of Language Models for Structure Prediction. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 803–823, Dublin, Ireland. Association for Computational Linguistics.

Yanbin Wei, Qiushi Huang, Yu Zhang, and James Kwok. 2023. KICGPT: Large Language Model with Knowledge in Context for Knowledge Graph Completion. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8667–8683, Singapore. Association for Computational Linguistics.

Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Shengping Liu, Bin Sun, Kang Liu, and Jun Zhao. 2023. Large Language Models are Better Reasoners with Self-Verification. *arXiv preprint*. ArXiv:2212.09561 [cs].

Xin Xu, Yuqi Zhu, Xiaohan Wang, and Ningyu Zhang. 2023. How to Unleash the Power of Large Language Models for Few-shot Relation Extraction?

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

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware Attention and Supervised Data Improve Slot Filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45, Copenhagen, Denmark. Association for Computational Linguistics.

Shaowen Zhou, Bowen Yu, Aixin Sun, Cheng Long, Jingyang Li, and Jian Sun. 2022. A survey on neural open information extraction: Current status and future directions. In *IJCAI*, pages 5694–5701. ijcai.org.

Yuqi Zhu, Xiaohan Wang, Jing Chen, Shuofei Qiao, Yixin Ou, Yunzhi Yao, Shumin Deng, Huajun Chen, and Ningyu Zhang. 2024. LLMs for Knowledge Graph Construction and Reasoning: Recent Capabilities and Future Opportunities. *arXiv preprint*. ArXiv:2305.13168 [cs].

A Implementation Details

In this section, we describe the specific methodologies and settings employed during our experiments to ensure clarity and reproducibility of the results. A unique separator token, "<==>", was utilized to facilitate the parsing of subjects and objects from the text. This token is not present in the original datasets, thereby avoiding any confusion with natural language text. Additionally, we inform LLMs about the task by starting our prompts with an instruction of the task. It is important to note that we have not conducted any prompt-tuning, as it is not the focus of this paper. Furthermore, we did not alter any hyperparameters related to the LLMs. The only hyperparameter that our framework includes is K, which represents the number of demonstrations in step 3.