Towards Harnessing Large Language Models as Autonomous Agents for Semantic Triple Extraction from Unstructured Text

Ananya Ananya[1,*], Sanju Tiwari[2], Nandana Mihindukulasooriya[3], Tommaso Soru[4], Ziwei Xu[5] and Diego Moussallem[6]

[1]*Indian Institute of Technology, Bhilai, India*
[2]*BVICAM, New Delhi, India & UAT, Mexico*
[3]*IBM Research, New York, United States*
[4]*Serendipity AI Ltd, London, United Kingdom*
[5]*National Institute of Advanced Industrial Science and Technology, Tokyo, Japan*
[6]*Jusbrasil, Salvador, Brazil*

Abstract

The use of Large Language Models as autonomous agents interacting with tools has shown to improve the performance of several tasks from code generation to API calling and sequencing. This paper proposes a framework for using Large Language Models as autonomous agents for the task of Knowledge Graph construction from unstructured text. Specifically, it focuses on triple extraction, which involves identifying entities and their relationships from text to construct a Knowledge Graph. Our novel framework "Auto-KG agent" incorporates two relation extraction tools, REBEL and KnowGL, in conjunction with Large Language Models. Experimental results on the CONLL04 dataset demonstrate that while multi-tool approaches face challenges like hallucination, LLM-based agents are promising in mitigating biases, major event identification, handling negations and modalities thus enhancing extraction accuracy, particularly for complex linguistic structures. The impetus for this research is to overcome the current limitations of existing systems for Knowledge Graph construction and propose a roadmap for developing a robust framework capable of handling the intricacies of natural language with minimal human interference. The paper also discusses future directions, such as emulating Large Language Model training using reinforcement learning with human feedback, incorporating query decomposition, and integrating a re-ranking module. Through this research, the authors aim to set a new direction for future endeavours in building advanced, reliable systems for knowledge extraction. Overall, this work highlights the potential of LLM-based agents for knowledge graph construction and proposes a framework for harnessing their capabilities.

Keywords

Triple extraction, Knowledge Graph, Knowledge Graph Construction, LLM Agents, Reasoning, Handling modalities and negations, Mitigating biases

TEXT2KG 2024: Third International Workshop on Knowledge Graph Generation from Text, May 26-30, 2024, co-located with Extended Semantic Web Conference (ESWC), Hersonissos, Greece
*Corresponding author.
ananyah@iitbhilai.ac.in (A. Ananya); tiwarisanju18@ieee.org (S. Tiwari); nandana@ibm.com (N. Mihindukulasooriya); tom@tommaso-soru.it (T. Soru); xuxiaowei23@hotmail.com (Z. Xu); diegomoussallem@gmail.com (D. Moussallem)
0009-0002-2431-2511 (A. Ananya); 0000-0003-1707-4842 (N. Mihindukulasooriya)

1. Introduction

The advent of Knowledge Graphs has revolutionized the way we represent and utilize information in the digital age. By structuring data as triples—consisting of a head entity, a relationship, and a tail entity (h, r, t) — Knowledge Graphs provide a semantic framework to describe the varied and countless entities and their interrelations in the objective world. This structured approach to data organization underpins intelligent applications and has garnered significant attention in both academic and industrial spheres due to its potent semantic processing capabilities and open organizational structure [1].

In the field of Natural Language Processing, extracting relational facts from text is crucial. Understanding the semantic relationships between entities in unstructured text helps convert raw data into structured formats. This structured data is extremely valuable for several tasks, such as building and enhancing Knowledge Bases. These bases are essential for powering applications that rely on knowledge [2].

In the realm of information extraction, frameworks like REBEL [2] and KnowGL [3] have emerged as powerful tools for converting unstructured text into structured relational data. These frameworks leverage the advancements in machine learning and natural language processing to perform tasks that traditionally required separate models for Named Entity Recognition (NER) and Relation Classification (RC). REBEL, which stands for Relation Extraction By End-to-end Language Generation, utilizes an autoregressive sequence-to-sequence (seq2seq) model, specifically a BART-large model, to extract relationships between entities in a text. The architecture of REBEL is designed to represent relations as a linearized sequence that includes entity mentions, labels, types, and the relation label. Similarly, KnowGL is a comprehensive framework that aims to transform natural language text into structured data that aligns with the schema of a Knowledge Graph like Wikidata. KnowGL consists of three main components: "Knowledge Generation", "Fact Ranking", and "Linking to Wikidata". We focus on the Knowledge Generation component of the KnowGL framework. The Knowledge Generation component uses fine-tuned, pre-trained language models to identify entity mentions and generate facts, including entity labels, types, and relationships.

Despite their innovative approaches, REBEL and KnowGL have certain limitations. The performance of these models is heavily influenced by the quality of their pre-training data. Biases or inaccuracies present in the training datasets can propagate through the models, affecting the accuracy of the extracted relations. Furthermore, the ability of these frameworks to generalize across various domains and text types hinges on the extent to which the pre-trained language models are fine-tuned or further pre-trained on domain-specific datasets. While Large Language Models inherently possess a broad understanding of language, their performance in specialized contexts improves significantly with targeted fine-tuning. Additionally, these systems may struggle with complex sentence structures and fail to identify all relevant major events, particularly in sentences laden with modalities or multiple clauses.

To address these challenges, we introduce a novel framework that synergizes the capabilities of REBEL and KnowGL with the nuanced understanding of Large Language Models. Large Language Models demonstrate remarkable efficacy in processing sentences with modalities and complex structures, leading to accurate event identification and triple extraction. This integration not only enhances event detection but also aids in mitigating biases inherent in

training data, ensuring a more comprehensive extraction of triples. Apart from introducing a novel framework, we also aim to answer the following research questions:

RQ1: How effective are Large Language Models in mitigating biases for extracting triples which are present in the datasets used for training information extraction tools?

RQ2: To what extent do Large Language Models accurately handle modalities and negations in natural language, and how does this capability affect the quality of triple extraction?

RQ3: Can Large Language Models enhance the identification of events within unstructured text, thereby improving the accuracy and completeness of triple extraction?

RQ4: How well do Large Language Models generalize across different datasets without the need for extensive training or fine-tuning, particularly in the context of triple extraction for knowledge graph construction?

RQ5: What is the impact of using multiple tools versus a single tool on the performance of triple extraction?

The impetus for our research is to overcome the current limitations of existing systems and chart a course for the development of a robust framework capable of handling the intricacies of natural language. Our experiments are designed with this objective in mind and are conducted with the resources available to us. Through this research, we aim to set a new direction for future endeavors in building advanced, reliable systems for knowledge extraction and reasoning.

2. Related Work

Using Large Language Models as autonomous agents has become increasingly popular in recent research. Large Language Models possess advanced reasoning abilities and skills in utilizing tools, making them well-suited for autonomous operations. They excel in tasks like acquiring knowledge, understanding instructions, generalizing information, planning, and reasoning, showcasing their potential for autonomous tasks [4]. However, Large Language Models do have limitations, such as performing arithmetic operations and staying updated with the latest information, which cannot be fully addressed through simple fine-tuning alone. This highlights the need for designing autonomous agent frameworks that can complement LLMs by integrating external data and supplementary tools [5].

This section has covered the existing studies on LLM-based Agents for Knowledge Graph Generation from Text. Jiang et. al [6] have introduced KG-Agent, an autonomous framework based on Large Language Models, designed to enable a small Large Language Model to independently make decisions throughout the reasoning process over Knowledge Graphs until completion. Within KG-Agent, a Large Language Model is combined with a versatile toolbox, a knowledge memory system and a KG-based executor. Jiang et. al [7] also introduced StructGPT tool, an Iterative Reading-then-Reasoning (IRR) framework aimed at addressing question-answering tasks using structured data. In this framework, a specialized interfaces has been designed to acquire relevant information from structured data and allowing Large Language Models to

focus on the reasoning tasks based on the acquired information. This research also introduced a procedure termed invoking linearization generation to promote Large Language Models in reasoning over structured data with the help of provided interfaces. Zhu et. al [8] explore Large Language Models for Knowledge Graph construction with reasoning and introduced an innovative approach called AutoKG, which utilizes multiple agents to efficiently handle both Knowledge Graph construction and reasoning tasks. There are several LLM-based Agents which are shown in Table 1.

Methodology	Description	Tasks	Dataset
Tool Learning [9]	In-context demonstration and Generation Regulation	Tool manipulation, multi-tool usage	ToolBench [9]
Instruction tuning [10]	Learning on high-quality instruction datasets	Tool manipulation, multi-tool usage	ToolBench [9]
Instruction Tuning with Human Curriculum [11]	Instruction data mimicking human learning progression	Reasoning and knowledge-based tasks	CORGI [11]
ReAct [12]	Prompting LLMs for Decision Making	Reasoning algorithm	HotPotQA [13], FEVER [14]
DFSDT [10]	Prompting LLMs for Decision Making	Reasoning algorithm	ToolBench [9]
CoT [15]	Prompting LLMs for Decision Making	Reasoning algorithm	GSM8K [16], SVAMP [17], ASDiv [18], AQuA, MAWPS [19]
AutoGPT [20]	Online Decision Making	Any decision-making task	ALFRED [21], DAgger [22]
WebGPT [23]	Text-based web browsing environment	Long-form QA	ELI5 [24]
MMREACT [25]	A system integrated with ChatGPT with a pool of vision experts	Multimodal reasoning and action	Self
ProgPrompt [26]	Prompt with program-like specifications of the available actions and objects in an environment	Generate situated Robot Task Plans	Self
LLM-ARK [27]	KG reasoning agent	Conversational Reasoning on Knowledge Graph, predictions on KG paths as a decision-making task	OpenDialKG [28]
KG-Agent [29]	Enables a small LLM to actively make decisions until finishing the reasoning process over KGs	Improve the reasoning ability and complex QA	WebQSP [30], CWQ [31], GrailQA [32]
KnowAgent [33]	Enhances the planning capabilities of LLMs by incorporating explicit action knowledge from KGs	A knowledgeable self-learning strategy for path planning	HotPotQA [13]

Table 1
A comparison of work which augments LLMs with tool usage

3. Background

In the construction of LLM agents, an LLM acts as the primary controller or "brain," orchestrating the sequence of operations required to accomplish a task or respond to a user query. These LLM agents may require additional modules like planning, memory, and tool utilization to enhance their functionality [34]. To activate the LLM component, a prompt template containing essential operational details and tool access specifications is utilized. While not mandatory, agents can be characterized or given a persona to define their role. This profiling information is typically embedded within the prompt and may include details such as role description, personality traits, social characteristics, and other demographic attributes [35].

Our research focuses on extracting triples from unstructured text to facilitate Knowledge Graph construction. Triple extraction entails identifying and extracting structured information in the form of triples, which comprise a subject entity, a relation or predicate, and an object

entity. This process is crucial in converting text into a structured format suitable for tasks like knowledge representation and information retrieval in natural language processing.

Relation extraction is another vital task we explore, involving the identification and extraction of semantic relationships between entities mentioned in unstructured text. These relationships, such as “is married to” or “works for”, capture meaningful associations and are represented as triples for downstream applications like Knowledge Graph construction.

REBEL [2] REBEL, which stands for Relation Extraction By End-to-end Language generation, is an extraction technique to pull out relationship details from raw text.

REBEL uses a special kind of model called autoregressive sequence-to-sequence (seq2seq) models. These models are good at making text and also understanding natural language. The main part of REBEL is a seq2seq model based on BART [36]. It represents relations between entities in the input text as a linearized sequence following a specific schema involving entity mentions, labels, types, and the relation label. REBEL uses BART-large as the base model which is first pre-trained on a large distantly supervised dataset called REBEL which was created by extracting over 800K training instances with 220 relation types from Wikipedia abstracts aligned with Wikidata facts. The pre-trained BART is then fine-tuned on this REBEL dataset to maximize the likelihood of generating the correct linearized triplet representation given the input text. REBEL demonstrates several advantages - it frames relation extraction in an end-to-end manner, can extract open-ended relation types, allows quickly fine-tuning on new datasets across domains, and achieves state-of-the-art performance on multiple relation extraction benchmarks while being simpler than prior complex pipeline approaches.

KnowGL [3] KnowGL is a comprehensive framework designed to convert natural language text into structured relational data that aligns with the schema of a Knowledge Graph like Wikidata. This framework comprises three key components: “Knowledge Generation”, “Fact Ranking”, and “Linking to Wikidata”. The Knowledge Generation aspect focuses on extracting facts by fine-tuning pre-trained language models to identify entity mentions and generate sets of facts including entity labels, types, and relationships. Fact Ranking involves parsing generated sequences to create a ranked list of distinct facts based on scores assigned to each fact. Lastly, Linking to Wikidata facilitates retrieving Wikidata IDs associated with the generated semantic annotations. By enabling the conversion of text into Wikidata statements in JSON format, KnowGL demonstrates the potential of pre-trained language models for generating structured data from text, offering an alternative to traditional information extraction pipelines.

4. System Architecture

The goal of our framework is to facilitate automatic triple extraction from text inputs. This framework is designed as a multi-tool system utilizing Large Language Models to execute the task of triple extraction.

Figure 1 outlines a framework for training a Large Language Model (LLM) using a method referred to as RLHF, which stands for Reinforcement Learning from Human Feedback. The flowchart is divided into two main sections:

1. The Large Language Model training procedure using RLHF [37]

2. RLHF LLM Based Autonomous Agents for triple Extraction for Knowledge Graph construction

LLM Training Procedure Using RLHF

The training procedure begins with raw text to pretrain the Large Language Model. This pre-training step is where the model learns from a large corpus of text to understand language patterns and structures.

After pretraining, the model becomes a "Pretrained LLM" which is then subjected to "Supervised Fine-tuning" using Demonstration data. Demonstration data consists of prompts and response pairs. This step involves training the model on specific examples to perform certain tasks and understand particular domains better. This form an instruction-following Chatbot of low quality.

The fine-tuned model, referred to as "SFT LLM" is then used in conjunction with "Human Preference Data" to train a "Reward Model". The Human Preference data consist of prompts, winner tuples and loser tuples. This reward model evaluates the outputs of the LLM and provides feedback on its performance.

The feedback from the reward model is used in Reinforcement Learning, where the Large Language Model is further trained using RLHF (Reinforcement Learning with Human Feedback) training prompts to improve its outputs based on human preferences and feedback. The RLHF training prompts also consists of prompt, winner tuples and loser tuples. In the end we get a high-quality instruction-following RLHF LLM [38]. The diagram presented in the first half of the Figure 1

RLHF LLM Based Autonomous Agents for Triple Extraction for KG Construction

The bottom section of the flowchart shows the application of the trained RLHF LLM for a specific task: Autonomous agents for triple extraction to construct Knowledge Graph.

A User inputs a complex query, a command which is then decomposed by the query-decomposition LLM. The decomposed query is processed by the "RLHF LLM" which interacts with a Tool DB (Tool Database). Here the tools in the Tool DB are "REBEL" and "KnowGL".

The RLHF LLM then performs re-ranking of triples extracted from the complex query which involves selecting the most relevant and accurate triples based on their count of occurrences of triples in the Knowledge Graph. The final output are then re-ranked which would be used in the construction of a Knowledge Graph. The system prompt for RLHF LLM is provided in the Appendix A.

It is a comprehensive framework for training a Large Language Model using human feedback and then applying this model to extract triples for building Knowledge Graph.

We implement the "Auto-KG Agent" framework to facilitate automatic triple extraction from text inputs. This framework is designed as multi-tool system utilizing Large Language Models to extract triples. We utilise REBEL and KnowGL as tools for triple extraction (relation along with entity). More such frameworks can be added as tools in the Tool DB. Large Language Model is asked to return the entities in JSON format. The diagram presented in the second half of the Figure 1 serves as the visual representation for this section.

The current system comprises the second half of the the Figure 1, without the query decomposition Large Language Model and Large Language Model training using RLHF. As of now we only incorporate Tool DB, LLM without RLHF (only system prompt) and re-ranking of triples

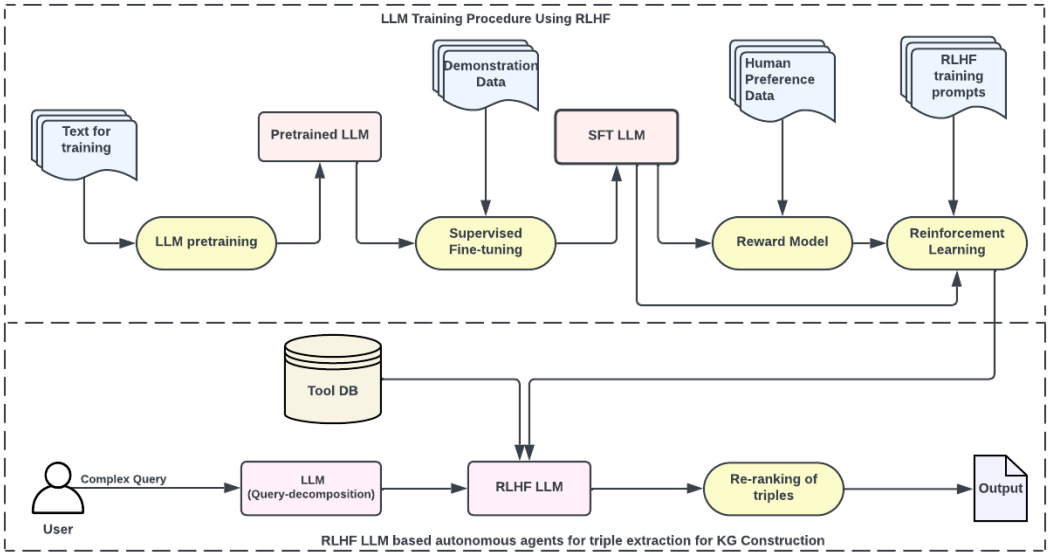

Figure 1: System architecture for utilising Large Language Models as autonomous agents enabling tools for Knowledge Graph from unstructured text

based on the length of relation extracted. We state in the section 7 for incorporating other modules in the "Auto-KG Agent" framework.

5. Preliminary Experimental Setup

Dataset We evaluate our system's performance on the CONLL04 dataset [39], which comprises sentences extracted from news articles. Each sentence is annotated with four entity types (person, organization, location, and other) and five relation types (kill, work for, organization based in, live in, and located in). Our evaluation focuses on the test split consisting of 288 instances [40], the ground truth, comparing the performance of our model against the REBEL model. The dataset statistics is described in Table 2.

Dataset	Entity Types	Relation Types	Train	Validation	Test
CONLL04	4	5	1,290 (922)	343 (231)	422 (288)

Table 2
Dataset statistics for CONLL04 (The number of instances are in brackets and the number of triples outside the bracket)

Evaluation Metrics The evaluation process compares the predicted triples extracted from test data with the ground truth triples. Each instance in both datasets is represented as a dictionary,

with a unique identifier and a set of triples. Each sentence has an object corresponding to it which stores the triple.

To calculate the true positives (correctly predicted triples), we iterate through each instance in the ground truth data. For each instance, we check if the corresponding instance exists in the predicted data. If it does, we find the intersection of the triples in the ground truth and predicted data, which gives us the number of correct predictions (true positives).

Additionally, we also calculate the number of extra predictions made by the model that is the count of triples not present in the ground truth. However, we don't calculate scores for them as it would require Human evaluation.

After calculating count of true positives, we compute both micro and macro scores for precision, recall, and F1 score. Micro scores consider the total number of triples in the entire dataset for calculating precision, recall, and F1 score, while macro scores average these metrics across each instance in the dataset.

Overall, this evaluation process enables us to assess the performance of the triple extraction model by quantifying its precision, recall, and F1 score, considering both individual instances and the entire dataset.

We did a strict evaluation where the correctness for triples extracted as a whole is compared with the corresponding head entity, tail entity and relation in the ground truth. Following are the counts of unique relations extracted for different frameworks:

- REBEL - 68 relations mapped to 5 relations
- KnowGL - 58 relations mapped to 5 relations
- REBEL + KnowGL - 90 relations mapped to 5 relations

Triple Extraction Tools REBEL and KnowGL frameworks are used as triple extraction tools in our experiment. In our evaluation of the REBEL model on the CONLL04 test dataset, we encountered a diverse set of 68 unique relations extracted by the model. To align these with the CONLL04 dataset's five predefined relations, we undertook a manual mapping process. This process was guided by semantic similarity and contextual relevance, ensuring that each extracted relation was correctly associated with one of the canonical relations such as 'killed by', 'residence', 'location', 'headquarters location', and 'employer', as originally formatted in the REBEL paper. The necessity for this manual mapping arose from the fact that the REBEL model, trained on multiple datasets, identified relations beyond the scope of the CONLL04 dataset, requiring careful consideration to maintain semantic integrity. The manual mappings for these relations, based on their semantic similarity to the corresponding five relations were carried out. Similarly, KnowGL had 54 unique relations being extracted. We followed the same mapping procedure for it. Figure 2 shows the distribution based on the number of occurrences for 5 types of relations extracted for different experiment design settings.

For details, readers can refer to Appendix A. The code for experiments is available in GitHub [1].

[1] https://github.com/Ananyaiitbhilai/Text2Triple-LLM-Agent

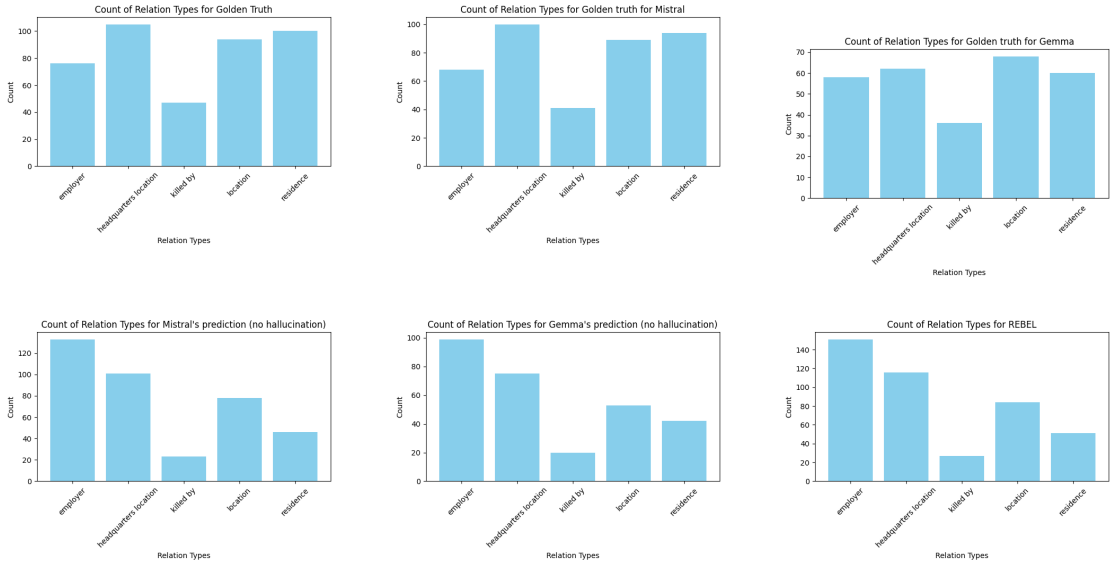

Figure 2: Count of Types of Relations for different Files

RLHF LLM In our experimental setup, we use Large Language Models without any fine-tuning or instruction tuning. However, in future we plan for RLHF LLMs being used to orchestrate the tool execution for triple extraction. Specifically, we used two open-source Large Language Models for benchmarking: "Gemma" and "Mistral"

Gemma [41] is a family of lightweight LLMs built from the same research and technology Google used to create the Gemini models. Gemma models are available in two sizes, 2 billion and 7 billion parameters. These models are trained on up to 6T tokens of primarily English web documents, mathematics, and code, using a transformer architecture with enhancements like Multi-Query Attention, RoPE Embeddings, GeGLU Activations, and advanced normalization techniques. We use the 2B one.

Mistral-7B-Instruct-v0.2 [42] Large Language Model (LLM) is an improved instruct fine-tuned version of Mistral-7B-Instruct-v0.1. The Mistral-7B-v0.1 Large Language Model is a pretrained generative text model with 7 billion parameters. Mistral-7B-v0.1 outperforms Llama 2 13B.

For additional experimental set-up refer Appendix 3.

The Table 3 illustrates the number of relations and instances for different experiment design settings.

In Table 4, REBEL refers to all triples extracted by REBEL, REBEL (subset Mistral) refers to removing all the triples which could not be extracted because of the hallucination (not returning triples in expected JSON format rather string format or other format) in the Mistral LLM. So those triples are removed from ground truth and then evaluation is carried out. Similar follows for REBEL (subset Gemma). Here the tools like KnowGL, REBEL are used as a single-tool in conjuction with LLM (Mistral and Gemma), REBEL and KnowGL as a multi-tool in conjuction with LLM (Mistral and Gemma).

Description	Number of Instances	Number of Relations
Full ground truth dataset with 5 relation types	288	5
Subset of ground truth for Gemma	187	5
Subset of ground truth for Mistral	262	5
Predictions extracted from Rebel	288	68
Predictions from Mistral	268	-
Predictions from Gemma	187	-
For comparison with Mistral	262	67
For comparison with Gemma	187	56
Predictions from Mistral without hallucination	262	66
Predictions from Gemma without hallucination	187	56
For comparison of REBEL with same subset as with Mistral, clustered	262	5
For comparison of REBEL with same subset as Gemma, clustered	187	5
Mistral predictions, clustered	262	5
Gemma predictions, clustered	187	5
Rebel predictions, clustered	288	5

Table 3
A summary of statistics for different experiment design settings.

Model	Micro			Macro			Extras
	P	R	F1	P	R	F1	
REBEL	0.16	0.16	0.16	0.18	0.18	0.18	356
REBEL (subset Mistral)	0.15	0.15	0.15	0.17	0.17	0.17	330
REBEL (subset Gemma)	0.18	0.19	0.18	0.21	0.21	0.21	235
MISTRAL (single-tool REBEL)	0.15	0.15	0.15	0.17	0.17	0.17	323
GEMMA (single-tool REBEL)	0.15	0.16	0.16	0.18	0.18	0.18	241
KNOWGL	0.05	0.05	0.05	0.07	0.06	0.07	372
Mistral (KNOWGL + REBEL multi-tool)	0.05	0.07	0.06	0.08	0.09	0.08	533

Table 4
Precision, Recall and F1 Scores (rounded upto 2 decimal places)

From Table 5 it can be observed that Gemma, a significant number of hallucinations were observed, accounting for 99 out of 288 total responses. This higher incidence of hallucination in Gemma was primarily attributed to incorrect JSON format returned by the model. Conversely, Mistral exhibited a lower occurrence of hallucination, with only 26 out of 288 total responses displaying such phenomena. The single-tool here refers to REBEL, and multi-tool has both REBEL and KnowGL. “Hallucination” refers to a phenomenon where the model generates text that is incorrect, nonsensical, or not real.

Model	Total Responses	Number of Hallucinations
Gemma	288	99
Mistral (single-tool)	288	26
Mistral (multi-tool)	288	42

Table 5
Comparison of Hallucination (particularly not giving response in expected JSON format) Occurrences

6. Results

We investigated the performance of multiple tools versus single tools for relation extraction and observed a notable decline in scores with multi-tool usage, suggesting that single-tool approaches may yield better results as shown in Table 4. We attributed this drop to increased hallucination, particularly more prevalent when employing multiple tools due to hallucination. However, single-tool usage also presented challenges, as occasionally, the returned format did not align with the one specified in the system prompt. Moreover, From Table 4 , it can be also be observed that REBEL with Large Language Model and only REBEL has almost same performance. It is due to the fact that REBEL and KnowGL is being used as a tool to trigger the action of extracting the relations. Both KnowGL and REBEL have the same architecture thus similar biases.

Our findings underscore the need for integrating Large Language Models with extraction tools to harness their full potential. While tools exhibits shortcomings in certain contexts, Large Language Models offer a complementary approach, particularly in mitigating biases and enhancing extraction accuracy. By empowering Large Language Models to engage in more nuanced planning or decomposing the query, we anticipate significant improvements in relation extraction performance.

Gemma's higher scores are because of a larger number of responses being generated as strings rather than in the expected JSON format as shown in Table 5. Consequently, these non-formatted responses are removed, resulting in fewer triples available for evaluation as compared to Mistral. This phenomenon contributes to Gemma's higher scores in triple extraction tasks compared to Mistral.

In order to answer our aforementioned Research Questions, we took certain examples that had multiple events, complex clauses, negation, modalities and then evaluation by human was carried out manually to check for the correctness of triples extracted from the pre-existing tools and our Auto-KG agent.

Event identification and mitigating biases in Triple Extraction Our investigation uncovered instances of flawed relation extraction within the REBEL and KnowGL tools. In a sample sentence, "While Marie Curie and Albert Einstein conducted groundbreaking experiments in their laboratories at the University of Paris, Leonardo da Vinci's sketches of Renaissance architecture in the bustling streets of Florence sparked inspiration across Italy," REBEL solely identified the triple "Florence, located in, Italy". However, this sentence encompasses two distinct events: "The experimentation conducted by Marie Curie and Albert Einstein at the University of Paris", and "the inspiration sparked by Leonardo da Vinci's sketches across Italy". REBEL's oversight stemmed from its inclination towards location-centric relations, influenced by biases within the training data.

In contrast, our Auto-KG agent showcased promise in mitigating such limitations. They accurately extracted triples from the sentence, capturing nuanced relations such as "Marie Curie, experimented at, University of Paris", "Albert Einstein, experimented at, University of Paris", "Leonardo da Vinci's sketches, located in, Florence", and "Leonardo da Vinci's sketches, sparked inspiration in, Italy". This underscores the Auto-KG agent's proficiency in comprehending complex linguistic structures and discerning meaningful relations, showcasing their potential for enhancing relation extraction accuracy.

Negation Handling Discrepancies in Triple Extraction In our comparative analysis, we also observed a notable discrepancy in the handling of negation between REBEL and KnowGL tools, as opposed to our Auto-KG agent in triple extraction tasks. Both REBEL and KnowGL demonstrated limitations in effectively managing negation cues within text, resulting in erroneous extraction of triples. Conversely, our tool exhibited robust performance in negation handling, yielding more accurate triple extractions even in the presence of negation cues.

We considered the sentence "Fado does not work at IIT". REBEL and KnowGL erroneously extract a triple indicating that Fado works at IIT, failing to account for the negation. In contrast, our tool was adept at discerning the negation cue "not" and appropriately adjusting the extracted triple to reflect the absence of the stated relationship, thereby accurately capturing the intended semantics of the sentence.

This discrepancy underscores the nuanced understanding of language exhibited by Large Language Models, enabling them to effectively navigate linguistic complexities such as negation cues in triple extraction tasks. It highlights the potential of leveraging LLM-based approaches to enhance the accuracy and reliability of triple extraction processes in natural language processing applications.

Generalising well on various datasets A significant disparity in the performance of seq2seq based approaches such as REBEL and KnowGL when trained or fine-tuned on specific datasets. While these models exhibit impressive performance within the confines of their training data, they demonstrate limited generalization capabilities beyond the dataset they were trained on. Conversely, Large Language Models showcase remarkable generalization prowess even without explicit training on a particular dataset. This discrepancy underscores the inherent adaptability and robustness of LLMs, enabling them to effectively handle diverse datasets and tasks without the need for extensive training or fine-tuning.

7. Future Directions

For future, our focus lies on emulating the Large Language Model training methodology using Reinforcement Learning with Human Feedback (RLHF) as detailed in Section 4

Additionally, we intend to incorporate a query-decomposition LLM to partition complex user queries into sub-queries, facilitating more precise event identification and subsequent triple extraction.

Furthermore, our proposed future work entails synergizing LLMs with multiple extraction tools to enhance their generalization capabilities across diverse datasets without requiring explicit training. This approach holds potential to surpass the performance of seq2seq models such as REBEL and KnowGL.

Moreover, we aim to integrate a re-ranking module into our framework. This module will prioritize all extracted triples based on their confidence levels, ensuring a more refined and accurate output.

We also aim to develop a diverse dataset that encompasses a wide range of relations and includes a variety of sentence structures. This dataset is intended to serve as a robust benchmark for evaluating performance in triple extraction tasks.

8. Limitations and Conclusion

The paper presents a novel framework that integrates Large Language Models (LLMs) with existing tools like REBEL and KnowGL for the task of triple extraction from unstructured text to construct knowledge graphs. The proposed framework aims to leverage the strengths of LLMs in understanding complex linguistic structures, handling modalities and negations, and mitigating biases inherent in training data. The experimental results on the CONLL04 dataset indicate that while multi-tool approaches face challenges such as hallucination, the integration of LLMs shows promising results in enhancing extraction accuracy.

There are certain limitation of our research work:

1. **Limited LLM Models Evaluated**: The experiments were confined to using the Gemma (2B parameters) and Mistral-7B LLMs. The performance of other large language models like LaMDA or models with higher parameter counts (e.g., GPT-4) remains unexplored. Future work could extend these experiments to a broader range of LLM architectures and sizes to provide a more comprehensive evaluation.

2. **Limited Task Coverage**: The current study focused on a specific task: triple extraction for knowledge graph construction. However, knowledge graph construction and reasoning encompass a wide range of tasks, and the performance of LLMs on other tasks, such as entity linking, relation classification, or multi-hop reasoning, remains unexplored. Future research could extend the evaluation to a broader set of tasks to provide a more comprehensive understanding of Large Language Model capabilities in the context of knowledge graph construction and reasoning.

3. **Limited Evaluation Dataset:** The paper evaluates the proposed framework on the CONLL04 dataset, which comprises sentences extracted from news articles with a limited set of entity types and relation types. This dataset may not fully represent the diversity and complexity of real-world text, potentially limiting the generalizability of the findings to other domains and contexts. In Future, evaluation can be carried out on other datasets

4. **Reliance on Manual Mapping:** The paper mentions that manual mapping was required to align the relations extracted by REBEL and KnowGL with the canonical relations in the CONLL04 dataset. This manual intervention introduces potential biases and inconsistencies, as the mapping process may not be entirely objective or scalable across larger datasets or domains.

The authors acknowledge these limitations and express their anticipation for future research opportunities that would allow them to further explore these areas and provide a more comprehensive evaluation of LLM capabilities in the context of knowledge graph construction and reasoning. The research sets a new direction for future work in building advanced, reliable systems for knowledge extraction and reasoning. It highlights the potential of LLM-based agents for knowledge graph construction and proposes a comprehensive framework for harnessing their capabilities.

References

[1] S. Jia, Y. Xiang, X. Chen, K. Wang, Triple trustworthiness measurement for knowledge graph, in: The World Wide Web Conference, WWW '19, ACM, 2019. URL: http://dx.doi.

org/10.1145/3308558.3313586. doi:10.1145/3308558.3313586.
[2] P.-L. Huguet Cabot, R. Navigli, REBEL: Relation extraction by end-to-end language generation, in: M.-F. Moens, X. Huang, L. Specia, S. W.-t. Yih (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2021, Association for Computational Linguistics, Punta Cana, Dominican Republic, 2021, pp. 2370–2381. URL: https://aclanthology.org/2021.findings-emnlp.204. doi:10.18653/v1/2021.findings-emnlp.204.
[3] G. Rossiello, M. F. M. Chowdhury, N. Mihindukulasooriya, O. Cornec, A. M. Gliozzo, Knowgl: Knowledge generation and linking from text, 2022. arXiv:2210.13952.
[4] L. Wang, C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang, Z. Chen, J. Tang, X. Chen, Y. Lin, W. Zhao, Z. Wei, J.-R. Wen, A survey on large language model based autonomous agents, 2023.
[5] H. Naveed, A. U. Khan, S. Qiu, M. Saqib, S. Anwar, M. Usman, N. Akhtar, N. Barnes, A. Mian, A comprehensive overview of large language models, 2024. arXiv:2307.06435.
[6] J. Jiang, K. Zhou, W. X. Zhao, Y. Song, C. Zhu, H. Zhu, J.-R. Wen, Kg-agent: An efficient autonomous agent framework for complex reasoning over knowledge graph, arXiv preprint arXiv:2402.11163 (2024).
[7] J. Jiang, K. Zhou, Z. Dong, K. Ye, W. X. Zhao, J.-R. Wen, Structgpt: A general framework for large language model to reason over structured data, arXiv preprint arXiv:2305.09645 (2023).
[8] Y. Zhu, X. Wang, J. Chen, S. Qiao, Y. Ou, Y. Yao, S. Deng, H. Chen, N. Zhang, Llms for knowledge graph construction and reasoning: Recent capabilities and future opportunities, arXiv preprint arXiv:2305.13168 (2023).
[9] Q. Xu, F. Hong, B. Li, C. Hu, Z. Chen, J. Zhang, On the tool manipulation capability of open-source large language models, 2023. arXiv:2305.16504.
[10] Y. Qin, S. Liang, Y. Ye, K. Zhu, L. Yan, Y. Lu, Y. Lin, X. Cong, X. Tang, B. Qian, S. Zhao, L. Hong, R. Tian, R. Xie, J. Zhou, M. Gerstein, D. Li, Z. Liu, M. Sun, Toolllm: Facilitating large language models to master 16000+ real-world apis, 2023. arXiv:2307.16789.
[11] B. W. Lee, H. Cho, K. M. Yoo, Instruction tuning with human curriculum, 2024. arXiv:2310.09518.
[12] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, Y. Cao, React: Synergizing reasoning and acting in language models, arXiv preprint arXiv:2210.03629 (2022).
[13] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. W. Cohen, R. Salakhutdinov, C. D. Manning, Hotpotqa: A dataset for diverse, explainable multi-hop question answering, 2018. arXiv:1809.09600.
[14] J. Thorne, A. Vlachos, C. Christodoulopoulos, A. Mittal, Fever: a large-scale dataset for fact extraction and verification, 2018. arXiv:1803.05355.
[15] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, D. Zhou, Chain-of-thought prompting elicits reasoning in large language models, 2023. arXiv:2201.11903.
[16] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, C. Hesse, J. Schulman, Training verifiers to solve math word problems, 2021. arXiv:2110.14168.
[17] A. Patel, S. Bhattamishra, N. Goyal, Are nlp models really able to solve simple math word problems?, 2021. arXiv:2103.07191.
[18] S.-y. Miao, C.-C. Liang, K.-Y. Su, A diverse corpus for evaluating and developing English math word problem solvers, in: D. Jurafsky, J. Chai, N. Schluter, J. Tetreault

(Eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 975–984. URL: https://aclanthology.org/2020.acl-main.92. doi:10.18653/v1/2020.acl-main.92.

[19] R. Koncel-Kedziorski, S. Roy, A. Amini, N. Kushman, H. Hajishirzi, MAWPS: A math word problem repository, in: K. Knight, A. Nenkova, O. Rambow (Eds.), Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, San Diego, California, 2016, pp. 1152–1157. URL: https://aclanthology.org/N16-1136. doi:10.18653/v1/N16-1136.

[20] H. Yang, S. Yue, Y. He, Auto-gpt for online decision making: Benchmarks and additional opinions, 2023. arXiv:2306.02224.

[21] M. Shridhar, J. Thomason, D. Gordon, Y. Bisk, W. Han, R. Mottaghi, L. Zettlemoyer, D. Fox, Alfred: A benchmark for interpreting grounded instructions for everyday tasks, 2020. arXiv:1912.01734.

[22] S. Ross, G. J. Gordon, J. A. Bagnell, A reduction of imitation learning and structured prediction to no-regret online learning, 2011. arXiv:1011.0686.

[23] R. Nakano, J. Hilton, S. Balaji, J. Wu, L. Ouyang, C. Kim, C. Hesse, S. Jain, V. Kosaraju, W. Saunders, X. Jiang, K. Cobbe, T. Eloundou, G. Krueger, K. Button, M. Knight, B. Chess, J. Schulman, Webgpt: Browser-assisted question-answering with human feedback, 2022. arXiv:2112.09332.

[24] A. Fan, Y. Jernite, E. Perez, D. Grangier, J. Weston, M. Auli, Eli5: Long form question answering, 2019. arXiv:1907.09190.

[25] Z. Yang, L. Li, J. Wang, K. Lin, E. Azarnasab, F. Ahmed, Z. Liu, C. Liu, M. Zeng, L. Wang, Mm-react: Prompting chatgpt for multimodal reasoning and action, 2023. arXiv:2303.11381.

[26] I. Singh, V. Blukis, A. Mousavian, A. Goyal, D. Xu, J. Tremblay, D. Fox, J. Thomason, A. Garg, Progprompt: Generating situated robot task plans using large language models, 2022. arXiv:2209.11302.

[27] Y. Huang, L. Shi, A. Liu, H. Xu, Evaluating and enhancing large language models for conversational reasoning on knowledge graphs, 2024. arXiv:2312.11282.

[28] S. Moon, P. Shah, A. Kumar, R. Subba, OpenDialKG: Explainable conversational reasoning with attention-based walks over knowledge graphs, in: A. Korhonen, D. Traum, L. Màrquez (Eds.), Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 845–854. URL: https://aclanthology.org/P19-1081. doi:10.18653/v1/P19-1081.

[29] J. Jiang, K. Zhou, W. X. Zhao, Y. Song, C. Zhu, H. Zhu, J.-R. Wen, Kg-agent: An efficient autonomous agent framework for complex reasoning over knowledge graph, 2024. arXiv:2402.11163.

[30] W. Yih, M. Richardson, C. Meek, M. Chang, J. Suh, The value of semantic parse labeling for knowledge base question answering, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers, The Association for Computer Linguistics, 2016.

[31] A. Talmor, J. Berant, The web as a knowledge-base for answering complex questions, in: M. A. Walker, H. Ji, A. Stent (Eds.), Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language

Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers), Association for Computational Linguistics, 2018, pp. 641–651.

[32] Y. Gu, S. Kase, M. Vanni, B. M. Sadler, P. Liang, X. Yan, Y. Su, Beyond I.I.D.: three levels of generalization for question answering on knowledge bases, in: J. Leskovec, M. Grobelnik, M. Najork, J. Tang, L. Zia (Eds.), WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021, ACM / IW3C2, 2021, pp. 3477–3488.

[33] Y. Zhu, S. Qiao, Y. Ou, S. Deng, N. Zhang, S. Lyu, Y. Shen, L. Liang, J. Gu, H. Chen, Knowagent: Knowledge-augmented planning for llm-based agents, arXiv preprint arXiv:2403.03101 (2024).

[34] E. Karpas, O. Abend, Y. Belinkov, B. Lenz, O. Lieber, N. Ratner, Y. Shoham, H. Bata, Y. Levine, K. Leyton-Brown, D. Muhlgay, N. Rozen, E. Schwartz, G. Shachaf, S. Shalev-Shwartz, A. Shashua, M. Tenenholtz, Mrkl systems: A modular, neuro-symbolic architecture that combines large language models, external knowledge sources and discrete reasoning, 2022. arXiv:2205.00445.

[35] L. Weng, Llm-powered autonomous agents, lilianweng.github.io (2023). URL: https://lilianweng.github.io/posts/2023-06-23-agent/.

[36] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer, Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, 2019. arXiv:1910.13461.

[37] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, T. Scialom, Llama 2: Open foundation and fine-tuned chat models, 2023. arXiv:2307.09288.

[38] M. Payne, Fine-tuning open llms with reinforcement learning from human feedback, https://www.width.ai/post/reinforcement-learning-from-human-feedback (2023). URL: https://www.width.ai/post/reinforcement-learning-from-human-feedback.

[39] D. Roth, W.-t. Yih, A linear programming formulation for global inference in natural language tasks, in: Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004, Association for Computational Linguistics, Boston, Massachusetts, USA, 2004, pp. 1–8. URL: https://aclanthology.org/W04-2401.

[40] P. Gupta, H. Schütze, B. Andrassy, Table filling multi-task recurrent neural network for joint entity and relation extraction, in: Y. Matsumoto, R. Prasad (Eds.), Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, The COLING 2016 Organizing Committee, Osaka, Japan, 2016, pp. 2537–2547. URL: https://aclanthology.org/C16-1239.

[41] G. Team, T. Mesnard, C. Hardin, R. Dadashi, S. Bhupatiraju, S. Pathak, L. Sifre, M. Rivière, M. S. Kale, J. Love, P. Tafti, L. Hussenot, A. Chowdhery, A. Roberts, A. Barua, A. Botev,

A. Castro-Ros, A. Slone, A. Héliou, A. Tacchetti, A. Bulanova, A. Paterson, B. Tsai, B. Shahriari, C. L. Lan, C. A. Choquette-Choo, C. Crepy, D. Cer, D. Ippolito, D. Reid, E. Buchatskaya, E. Ni, E. Noland, G. Yan, G. Tucker, G.-C. Muraru, G. Rozhdestvenskiy, H. Michalewski, I. Tenney, I. Grishchenko, J. Austin, J. Keeling, J. Labanowski, J.-B. Lespiau, J. Stanway, J. Brennan, J. Chen, J. Ferret, J. Chiu, J. Mao-Jones, K. Lee, K. Yu, K. Millican, L. L. Sjoesund, L. Lee, L. Dixon, M. Reid, M. Mikuła, M. Wirth, M. Sharman, N. Chinaev, N. Thain, O. Bachem, O. Chang, O. Wahltinez, P. Bailey, P. Michel, P. Yotov, P. G. Sessa, R. Chaabouni, R. Comanescu, R. Jana, R. Anil, R. McIlroy, R. Liu, R. Mullins, S. L. Smith, S. Borgeaud, S. Girgin, S. Douglas, S. Pandya, S. Shakeri, S. De, T. Klimenko, T. Hennigan, V. Feinberg, W. Stokowiec, Y. hui Chen, Z. Ahmed, Z. Gong, T. Warkentin, L. Peran, M. Giang, C. Farabet, O. Vinyals, J. Dean, K. Kavukcuoglu, D. Hassabis, Z. Ghahramani, D. Eck, J. Barral, F. Pereira, E. Collins, A. Joulin, N. Fiedel, E. Senter, A. Andreev, K. Kenealy, Gemma: Open models based on gemini research and technology, 2024. arXiv:2403.08295.

[42] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, W. E. Sayed, Mistral 7b, 2023. arXiv:2310.06825.

A. Appendix: Additional Details

Additional details about System and parameters for the preliminary Experimental set-up

- The context size(n_ctx) is the maximum number of tokens that the model can account for when processing a response. this includes the prompt, and the response itself. In our case the context size was set to 2048.
- The maximum number of tokens to generate is 2000 in our case. If *max_tokens* ≤ 0 or None, the maximum number of tokens to generate is unlimited and depends on n_ctx.
- Average inference time per context/sentence for CONLL04 test dataset for extracting triples in conjunction with LLMs was 25 seconds.
- Temperature was set to 0
- The gguf files for Mistral and Gemma were run locally on Mac M1.

The system prompt is shown in Figure 3

Relation Mappings

Key	Values

employer	derivative work, inception, instance of, owned by, owner of, part of, participant, participant in, performer, twinned administrative body, occupation, field of this occupation, member of political party, work location, language used, participant in, participant, owner of, owned by, member of, notable work, instance of, interested in, office held by head of government, chief executive officer, educated at, subclass of, part of, office held by head of state, chairperson, executive body, industry, officeholder, position held, practiced by, language of work or name, director / manager, employer, field of work, language of work or name, notable work, occupation, member of, member of political party, officeholder, operator, position held, educated at, founded by, product or material produced, subsidiary, work location, author, office held by head of government, used by, uses, candidacy in election, candidate, chairperson, head of government
headquarters location	headquarters location, twinned administrative body, applies to jurisdiction, legislative body, military branch, contains administrative territorial entity, parent organization, operating area, legislative body, contains administrative territorial entity, headquarters location, located in the administrative territorial entity, ethnic group, language used, military branch, parent organization, applies to jurisdiction
killed by	cause of death, perpetrator, convicted of, killed by, place of death, facet of, date of death, main subject, place of death, facet of, significant event
location	location, capital, continent, located in time zone, shares border with, mountain range, located in or next to body of water, candidate, significant place, spouse, place of publication, target, country, located in or next to body of water, location, mouth of the watercourse, point in time, capital, capital of, shares border with, tributary, diplomatic relation, place of publication, spouse
residence	place of birth, based on, country of citizenship, date of birth, has part, number of participants, history of topic, place of birth, country of origin, has quality, significant event, occupant, relative, residence

System Prompt

```
'''<s>[INST] <<SYS>>
Assistant is an expert JSON builder designed to assist with a wide range of tasks.

Assistant is able to trigger actions for User by responding with JSON strings that contain "action" and
"action_input" parameters.

The available actions to Assistant are:

- "extract_text_triplets": Useful when Assistant is asked to extract triplets from a given text.
  - To use the extract_triplets tool, Assistant should respond like so:
    {{"action": "extract_text_triplets", "action_input": "Your text here"}}

- "extract_entities": Useful when Assistant is asked to extract entities from a given text.
  - To use the extract_triplets tool, Assistant should respond like so:
    {{"action": "extract_entities", "action_input": "Your text here"}}

Here are some previous conversations between the Assistant and User:

User: Hey how are you today?
Assistant: I'm good thanks, how are you?
User: Can you extract all the triplets from this text: "Gràcia is a district of the city of Barcelona, Spain."
Assistant: {{"action": "extract_text_triplets", "action_input": "Gràcia is a district of the city of Barcelona,
Spain."}}
User: Also give triples for "obama was US president"
Assistant: {{"action": "extract_text_triplets", "action_input": "obama was US president"}}
User: Can you extract all the entities from this text: "Gràcia is a district of the city of Barcelona, Spain."
Assistant: {{"action": "extract_entities", "action_input": "Gràcia is a district of the city of Barcelona, Spain."}}
User: Also give entities for "obama was US president"
Assistant: {{"action": "extract_entities", "action_input": "obama was US president"}}


<</SYS>>

{0}[/INST]'''
```

Figure 3: System Prompt