Text2KGBench: A Benchmark for Ontology-Driven Knowledge Graph Generation from Text

Nandana Mihindukulasooriya[1](✉), Sanju Tiwari[2], Carlos F. Enguix[2], and Kusum Lata[3]

[1] IBM Research Europe, Dublin, Ireland
nandana@ibm.com
[2] Universidad Autonoma de Tamaulipas, Victoria, Mexico
[3] Sharda University, Greater Noida, India

Abstract. The recent advances in large language models (LLM) and foundation models with emergent capabilities have been shown to improve the performance of many NLP tasks. LLMs and Knowledge Graphs (KG) can complement each other such that LLMs can be used for KG construction or completion while existing KGs can be used for different tasks such as making LLM outputs explainable or fact-checking in Neuro-Symbolic manner. In this paper, we present *Text2KGBench*, a benchmark to evaluate the capabilities of language models to generate KGs from natural language text guided by an ontology. Given an input ontology and a set of sentences, the task is to extract facts from the text while complying with the given ontology (concepts, relations, domain/range constraints) and being faithful to the input sentences. We provide two datasets (i) Wikidata-TekGen with 10 ontologies and 13,474 sentences and (ii) DBpedia-WebNLG with 19 ontologies and 4,860 sentences. We define seven evaluation metrics to measure fact extraction performance, ontology conformance, and hallucinations by LLMs. Furthermore, we provide results for two baseline models, Vicuna-13B and Alpaca-LoRA-13B using automatic prompt generation from test cases. The baseline results show that there is room for improvement using both Semantic Web and Natural Language Processing techniques.

Resource Type: Evaluation Benchmark
Source Repo: https://github.com/cenguix/Text2KGBench
DOI: https://doi.org/10.5281/zenodo.7916716
License: Creative Commons Attribution (CC BY 4.0)

Keywords: Benchmark · Relation Extraction · Knowledge Graph · Knowledge Graph Generation · Large Language Models

1 Introduction

Knowledge Graphs (KG) are becoming popular in both industry and academia due to their useful applications in a wide range of tasks such as question answering, recommendations, semantic search, and advanced analytics with explainability [16].

T. R. Payne et al. (Eds.): ISWC 2023, LNCS 14266, pp. 247–265, 2023.
https://doi.org/10.1007/978-3-031-47243-5_14

A KG can be generated using mappings such as RDB2RDF [38] if the source is relational data or semi-structured using RML [11]. Crowdsourcing can be used to build them manually as in Wikidata [48]. However, there are cases where the data is in unstructured format in text documents and crowd-sourcing is not an option (for example, internal documents). One solution in such cases is to construct knowledge graphs using Natural Language Processing (NLP) techniques such as Named Entity Recognition (NER), Relation Extraction, Open Information Extraction, Entity Linking, and Relation Linking. There is a growing interest in the Semantic Web community to explore such approaches as seen from the workshops such as Text2KG [43,44] and NLP4KGC [46].

The recent advances in large language models (LLM) and foundation models with emergent capabilities have been shown to improve the performance in many NLP tasks [6]. KGs and LLMs can complement each other in both directions; on the one hand, LLMs can be helpful in constructing KGs and on the other hand KGs can be used to validate LLM outputs or make them explainable. Approaches such as Neuro-Symbolic AI [15] will allow using KGs and LLMs jointly. In order to foment research in this direction, the establishment of evaluation benchmarks is necessary. In this context, *Text2KGBench* is a benchmark for measuring the capabilities of LLMs for generating KGs from text conforming to a given ontology. In this version, we are not evaluating the ability to process or generate RDF/OWL representations but rather the ability of extracting facts using correct relations.

There are several manners LLMs can be adapted to this task, including fine tuning [17] (also known as model tuning), updating all model parameters, Prompt tuning [24] or Prefix-Tuning [26] by keeping the model parameters frozen and only prefixing some tunable tokens to the input text and prompt design where the model is used as it is, but the prompt or the input to the model is designed to provide a few examples of the task [6]. Each of these approaches has their pros and cons with respect to the performance, computation resources, training time, domain adaption and training data required. Our benchmark provides training data that can be used in any of those approaches.

In-context learning [31,51] with prompt design is about teaching a model to perform a new task only by providing a few demonstrations of input-output pairs at inference time. Instruction fine-tuning using approaches such as InstructGPT [34], Reinforcement Learning from Human Feedback (RLHF) [9,41] significantly improves the models capabilities to follow a broad range of written instructions.

A vast number of LLMs have been released in recent months [52], especially in the GPT family of models such as GPT-3 [6], ChatGPT, LLaMA [45], BLOOM [39], PaLM [8], and Bard. Such models can be easily adapted for KG generation from text with a prompt design containing instructions and contextual information.

The main contributions of this paper are:

- We propose a novel benchmark *Text2KGBench* by extending the relation extraction by guiding it with ontology and instructions. We provide two datasets, (a) Wikidata-TekGen with 10 ontologies and 13,474 sentences aligned to triples and (b) DBpedia-WebNLG with 19 ontologies and 4,860 sentences aligned to triples by reusing TekGen [1] and WebNLG [13] corpora. We define seven metrics for measuring the accuracy of fact extraction, ontology conformance and detecting hallucinations and provide evaluation scripts.

- We provide results for two baselines using open-source LLMs, including Vicuna-13B [7] and Alpaca-LoRA-13B [19,42] with in-context learning. We also provide a baseline automatic prompt generator from ontologies and approach finding best demonstration examples with sentence similarity using SBERT T5-XXL model [33,36]. We provide all generated prompts, similarities, and LLM responses for further analysis.

The rest of the paper is organized as follows. Section 2 introduces the task of the benchmark, Sect. 3 describes how the benchmark was created, Sect. 4 defines the evaluation metrics and Sect. 5 presents the baselines and evaluation results. After related work in Sect. 6, the paper concludes with some final remarks and future work in Sect. 7.

2 Task Description

This section introduces the task of *Text2KGBench*. With the recent advancements of LLMs, we envision that LLMs can be used to generate KGs guided by ontologies as illustrated in Fig. 1. Given an ontology and text corpora, the goal is to construct prompts to instruct the model to extract facts relevant to the ontology. Such extracted facts can be further validated and post-processed to create a knowledge graph.

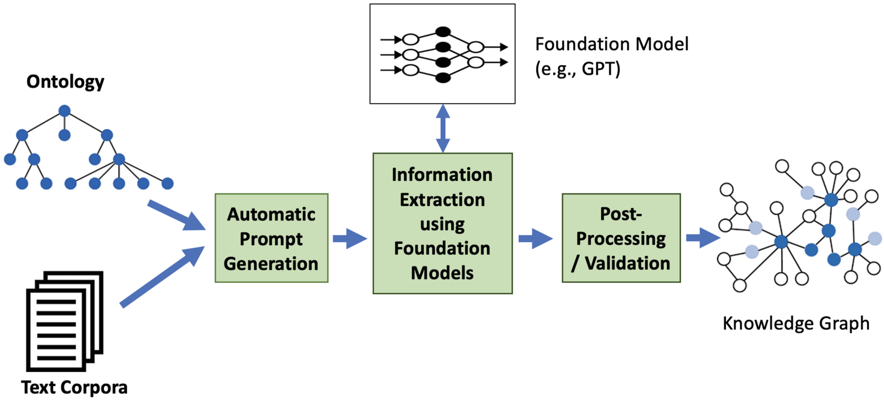

Foundation Model
(e.g., GPT)
Ontology
Information Extraction using Foundation Models
Automatic Prompt Generation
Post-Processing / Validation
Knowledge Graph
Text Corpora

Fig. 1. Generating knowledge graphs from text guided by ontologies

In the context of the *Text2KGBench*, we define the task as a fact extraction task guided by an ontology. The proposed task is closely related to the relation extraction and relation classification tasks in literature but with an explicit ontology definition given as input. There are three main inputs to the task:

Ontology: The ontology defines the concepts of interest, a set of defined relations with their canonical names, domain and range constraints for the relations. This can be further extended with other ontological axioms to guide models.

Text Corpus: The text corpus contains the set of natural language sentences that contains facts that can be expressed using the aforementioned ontology.

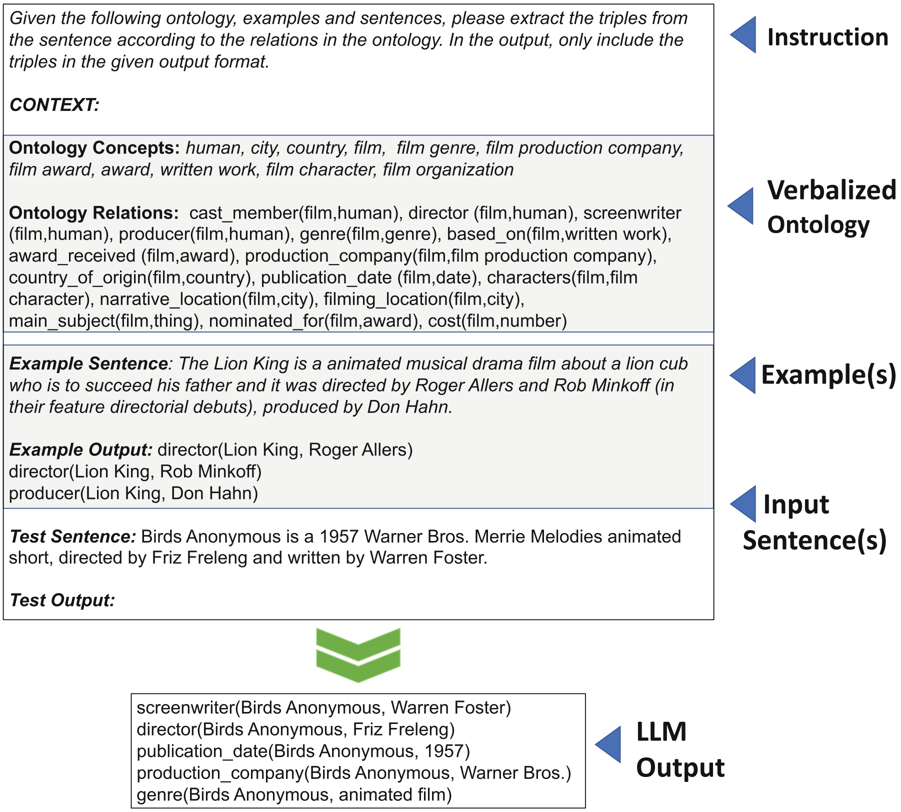
Examples: Demonstrative examples or training data contains pairs of sentences and the facts extracted from them complying with the ontology.

Given these inputs, a system should be able to generate facts adhering to a set of expectations. First, the system should use the ontology and demonstrative examples as guidance on which facts to extract and which relations to be used in the output. It should follow the canonical relation names and the example output format. In the evaluation, we measure this aspect using ontology compliance metrics. Second, the system should be faithful to the input sentence. This means the system should consider only the facts mentioned in the sentence as the truth (irrespective of the knowledge it may have from pre-training). It should not include additional information that is not directly or indirectly stated or implied by the sentence. This aspect is measured by the fact extraction accuracy metrics. Finally, the system should not hallucinate i.e. it should not introduce new or fake entities/relations not mentioned in the sentence and the ontology. This aspect is measured by the hallucination metrics. Section 4 provides details of evaluation metrics.

In this version of *Text2KGBench*, we are not evaluating a system's ability to process RDF/OWL syntax or a deep understanding of OWL semantics. Thus, we are using simpler language-oriented verbalizations and triple formats for presenting the information to an LLM. Figure 2 illustrates an example of performing the task using in-context learning of LLMs with a prompt.

There are several components or lines of research that can affect the results of a system tested under this benchmark. One of the most important aspects is the model (LLM) being used. Depending on the characteristics such as the architecture, training data being used, number of parameters, and what instructions have been used for fine-tuning, each of the language models can have different capabilities, and it has a direct impact on the results obtained from the model.

Prompt engineering or automatic prompt generation also plays a vital role in this task. Recently, there is a line of research that is focused on how to build efficient prompts for getting expected outputs from LLMs. In this benchmark, the participants can design different prompts guided by an ontology and reasoning techniques can be used to develop the most efficient prompts. Related to prompt generation, another important aspect is how to find the most relevant or helpful demonstration example from training data given a test case. This can be done using sentence similarity metrics or utilizing more advanced semantic clues from the ontology.



Given the following ontology, examples and sentences, please extract the triples from the sentence according to the relations in the ontology. In the output, only include the triples in the given output format.

CONTEXT:

Ontology Concepts: *human, city, country, film, film genre, film production company, film award, award, written work, film character, film organization*

Ontology Relations: cast_member(film,human), director (film,human), screenwriter (film,human), producer(film,human), genre(film,genre), based_on(film,written work), award_received (film,award), production_company(film,film production company), country_of_origin(film,country), publication_date (film,date), characters(film,film character), narrative_location(film,city), filming_location(film,city), main_subject(film,thing), nominated_for(film,award), cost(film,number)

***Example Sentence**: The Lion King is a animated musical drama film about a lion cub who is to succeed his father and it was directed by Roger Allers and Rob Minkoff (in their feature directorial debuts), produced by Don Hahn.*

Example Output: director(Lion King, Roger Allers)
director(Lion King, Rob Minkoff)
producer(Lion King, Don Hahn)

Test Sentence: Birds Anonymous is a 1957 Warner Bros. Merrie Melodies animated short, directed by Friz Freleng and written by Warren Foster.

Test Output:

screenwriter(Birds Anonymous, Warren Foster)
director(Birds Anonymous, Friz Freleng)
publication_date(Birds Anonymous, 1957)
production_company(Birds Anonymous, Warner Bros.)
genre(Birds Anonymous, animated film)

Fig. 2. An example prompt for an instruction fine-tuned LLM and the generated output from the LLM model.

Post-processing and validation are also crucial for extracting the correct triples and cleaning them by removing implausible triples. Initial extraction can be done using pattern-matching techniques such as using regex. Validation of the generated triples is another open research area which can use linguistic approaches to detect hallucinations and reasoning-based approaches to validate that the generated triples are consistent with the ontology.

3 Benchmark Generation

Text2KGBench consists of two datasets: *wikidata-tekgen* and *dbpedia-webnlg*. As discussed above, each of those has a set of ontologies and corpora of text where sentences are aligned with triples according to the given ontology.

3.1 Wikidata-TekGen Dataset

This dataset is created using sentence alignments provided by TekGen corpus.

Ontology Selection. As the first step of building the dataset, we have created 10 small ontologies by reusing the concepts and relations described in Wikidata. We selected a domain, such as movies or sports and explored the concepts and relations relevant to the given domain in Wikidata. With that, a set of concepts for the domain are identified, and a sample of their instances is checked for the most frequent relations. Once a relation is identified, it's property page is used to understand the usage, and domain range constraints. For example, the property page[1] for the relation "director (P57)" describes subject and value-type constraints. Iteratively, more concepts are added to the ontology based on the domain/range constraints of the selected relations. This process is performed manually and each ontology was formulated by an author with Semantic Web expertise and reviewed by two other experts. Table 1 shows the concept and relation statistics for each of the 10 ontologies that we generated.

An example ontology for the music domain is shown in Fig. 3. All 10 ontologies are available as OWL ontologies serialized in Turtle and in a compact json format in the repo[2].

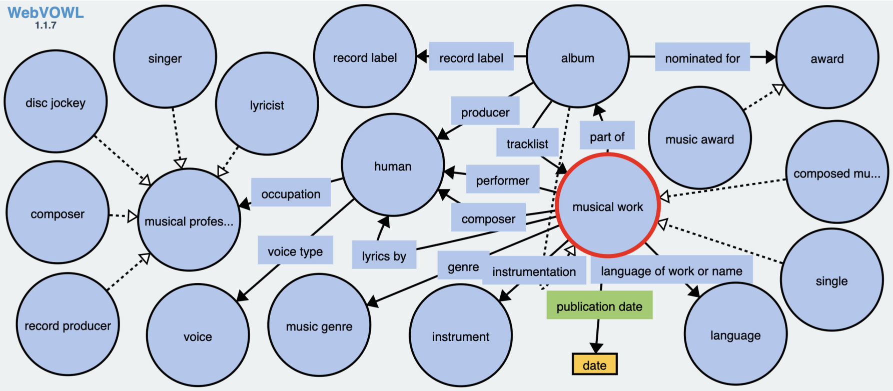


Fig. 3. An illustration of the music ontology with concepts and relations selected from Wikidata.

Triple Generation and Alignment with Sentences. Given an ontology from the previous step, a parameterized SPARQL query[3] is used to generate a set of K triples for each of the relations. The SPARQL query guaranteed that the triples confirmed the domain and range restrictions of each ontology. For example, for "director" relation, we would get triples such as director("Lion King","Roger Allers").

[1] https://www.wikidata.org/wiki/Property:P57.

[2] https://github.com/cenguix/Text2KGBench/tree/main/data/wikidata_tekgen/ontologies.

[3] https://github.com/cenguix/Text2KGBench/tree/main/src/benchmark.

In this dataset, we reused the TekGen corpus [1] which provides Wikidata triples aligned with corresponding sentences from Wikipedia. The TekGen corpus is generated using distant supervision and it has 16 M aligned triple-sentences covering 663 Wikidata relations. For each triple we got from the previous step, we analyzed the TekGen corpus to get an aligned sentence when available. For instance, the triple in the previous sentence will be aligned to a sentence such as "The Lion King is an animated musical drama film directed by Roger Allers and Rob Minkoff, produced by Don Hahn.“. Once a sentence is found, we check all the other relations associated with the sentence in the TekGen corpus and include them also if they are part of our ontology. For example, in this sentence, director ("Lion King", "Rob Minkoff") and producer("Lion King", "Don Hahn") will also be included in the dataset.

Once we complete this process for all 10 ontologies, we generated 13,474 sentence - triple(s) alignments and they are divided into train, validation and test sets.

Manual Validations and Cleaning. Because the TekGen corpus is generated using distant supervision, it can have noise and some incorrect alignments. In order to evaluate models with a more precise set of test cases, we have manually analyzed the test sentences and selected a smaller subset of more accurately aligned sentences for each ontology. For this exercise, the annotators looked at the triple and aligned sentence in the gold standard and selected sentences that a human can easily extract the triple such that the fact is explicitly mentioned in the text. For example, "The film was also nominated for Academy Award for Best Picture." is a noisy sentence to extract the triple "nominated for(Working Girl, Academy Award for Best Picture) as it is impossible for a model to resolve coreference to understand what term "the film" is referring to, only with this sentence as input. Another example, the sentence "Welcome to Eltingville was written by Dorkin and Chuck Sheetz" is wrongly aligned with the triple director("Welcome to Eltingville", "Chuck Sheetz") because the entities co-occur in the sentence and Chuck Sheetz is both the director and the writer. For a sample of test data, the authors removed such alignments and created another test set with 939 verified sentence-triple alignments. The systems can use both the larger test set and this smaller high-quality test set for their evaluations.

Unseen Sentence Generation. One of the caveats of this benchmark is that the language models under test might have already seen these sentences or even the alignments in some form. Then it can be argued that they might have memorized some of these relations. One important aspect to evaluate is if the model performance will get affected if we test the model with unseen sentences that are not part of Wikipedia and not seen during the pre-training. For that, we invent new sentences with facts that the annotators come up with. For example, a sentence such as "John Doe starred in the movie The Fake Movie released in 2025". With this exercise, the authors generated 174 unseen sentences roughly two sentences per each relation in each ontology. Furthermore, this unseen set of sentences can be used to check how faithful the model is to the given sentence when generating the triples.

Table 1. Statistics related to the two datasets including the list of ontologies, number of types and relations in each ontology, and number of sentences aligned.

wikidata-tekgen				dbpedia-webnlg							
Ontology	Types	Rels.	Sents.	Ontology	Types	Rels	Sents.	Ontology	Types	Rels	Sents.
Movie	12	15	2800	University	15	46	156	Transport	20	68	314
Music	13	13	2243	Music	15	35	290	Monument	14	26	92
Sport	15	11	1693	Airport	14	39	306	Food	12	24	398
Book	20	12	1810	Building	14	38	275	Written Work	10	44	322
Military	13	9	750	Athlete	17	37	293	Sports Team	14	24	235
Computer	15	12	743	Politician	19	40	319	City	11	23	348
Space	15	7	666	Company	10	28	153	Artist	20	39	386
Politics	13	9	695	Celestial	8	27	194	Scientist	15	47	259
Nature	14	13	1558	Astronaut	16	38	154	Film	18	44	264
Culture	15	8	516	Comics	10	18	102				
Total			13,474	Total							4,860

3.2 DBpedia-WebNLG Dataset

The DBpedia-WebNLG dataset is created reusing the alignments in the WebNLG corpus.

Ontology Selection. Similar to the previous dataset, the first step is to create a set of ontologies. WebNLG consists of 19 categories and we created an ontology for each category. First, we analysed the triples in each category to extract the relations in each category and defined the concepts based on the domain and range constraints of those relations. The statistics for the resulting 19 ontologies are shown in Table 1.

Triple Generation and Alignment with Sentences. We have parsed the WebNLG 3.0 English dataset and collected the sentences in one of the splits (WebNLG 3triples). When creating train and test sets, we made sure that the same fact would not appear in both train and test sets. Because the alignments (verbalizations) are verified by crowdsourcing in WebNLG, there was no need for us to create a manually validated set. We generated 4,860 sentence - triple(s) alignments using WebNLG data and divided into train and test splits.

The train/val/test splits for both benchmarks were done as stratified randomized folds aiming to preserve the relation distributions as much as possible using scikit-learn. The rationale for the splits was to provide training data (examples for in-context learning or fine-tuning models) for future systems that will use the benchmark and validation data (for optimizing hyperparameters).

4 Evaluation Metrics

In this section, we present the set of evaluation metrics we use in *Text2KGBench* to measure the performance of systems for generating facts from the text. Evaluation metrics aim to validate three aspects: (i) extracted facts are accurate according to the given ontology, (ii) extracted facts conform to the given ontology, and (iii) the output doesn't include any hallucinations.

Given a prompt similar to Fig. 2, the LLM will produce a textual output that can be parsed into a set of triples, which we call LLM output triples. The expected output for each test sentence is in the ground truth files.

Fact Extraction Accuracy: This is measured using Precision (P), Recall (R), and F1 scores by comparing LLM output triples to the ground truth triples. P is calculated by dividing the number of correct LLM triples (which are part of the ground truth) by the number of LLM triples. R is calculated by dividing the number of correct LLM triples by the number of ground truth triples. F1 is calculated as the harmonic mean of the P and R. If the LLM output is empty, P, R, and F1 are set to 0. Because the set of triples are not exhaustive for a given sentence, to avoid false negatives, we follow a locally closed approach by only considering the relations that are part of the ground truth. For P, R, F1, higher numbers represent better performance.

Ontology Conformance: This is measured using the Ontology Conformance (OC) metric which is calculated as the percentage of LLM output triples conforming to the input ontology, i.e., ontology conforming LLM output triples divided by total LLM output triples. In this version, a triple is considered to be conforming to the ontology if the relation is one of the canonical relations listed in the ontology. This can be further extended to validate other restrictions such as domain, range or other ontological axioms.

Hallucinations: Hallucination is defined as the generated content that is nonsensical or unfaithful to the provided source content [21]. We calculate three hallucination metrics, subject hallucination (SH), relation hallucination (RH), and object hallucination (OH). These are calculated by comparing the generated triple to the test sentence and the ontology. For each triple, SH and OH check if the subject and object are present in either the sentence or the ontology concepts, and RH checks if the relation is present in the ontology relations. For SH and OH, we use stemming to account for inflected forms with morphological variations such as “America”, “American”, “Americans”, etc. Each term in the subject or object and test sentence is stemmed before checking if the subject or object is present as a substring in the test sentence and/or ontology concepts. For RH, relations are matched using exact matches. In this version, RH and OC are inversely related *i.e.* 1 - OC equals to RH.

5 Baselines and Evaluation Results

In this section, we present baseline LLM models, baselines for automatic prompt generation and the evaluation results for baseline models for the two datasets of *Text2KGBench* we described in Sect. 3.

5.1 Baseline LLM Models

Vicuna-13B. Vicuna-13B [7] is an open-source LLM that fine-tunes a base LLaMA model with 70K user-shared conversations from ShareGPT. We obtained the LlaMA 13B LLM model, checkpoints, and tokenizer, through the Pyllama Github repository[4] and applied Vicuna weights from FastChat[5] as delta weights. Vicuna-13B claims 90% performance of OpenAI ChatGPT and Google Bard [56] where the authors have used a metric "Relative Response Quality" using strong LLM (GPT4) as judges to evaluate the model on open-ended questions.

Alpaca-LoRA-13B. Alpaca-LoRA[6] is a model that fine-tuned a base LLaMA model with the same 52K instructions of Alpaca model that is generated using self-instruct [50] with the OpenAI's *text-davinci-003* model. Alpaca-LoRA is fine-tuned using Low-Rank Adaptation [19] allows reducing the number of trainable parameters by a significant order by freezing the pre-trained model weights and injecting trainable rank decomposition matrices to each transformer layer.

5.2 Automatic Prompt Generation

Both our LLM models are GPT-style decoder-only models that are instruction fine-tuned. They can be used for downstream tasks by providing a prompt with an instruction. In this section, we present the steps involved in automatically creating prompts for each test sentence.

Our baseline prompt consists of our main parts: (a) Instruction, (b) Ontology description, (c) Demonstrative examples, and (d) Test sentence as illustrated in Fig. 2.

Instruction. This is a fixed instruction that we used for all test cases across the ontologies. We used the following phrase "Given the following ontology and sentences, please extract the triples from the sentence according to the relations in the ontology. In the output, only include the triples in the given output format." as the instruction. We describe the task as well as request the model to be less verbose and output only the triples in the given format.

Ontology Description. This part of the prompt provides a description of the ontology to the model as context. Each test case in our benchmark is associated with an ontology. This part of the prompts verbalizes the ontology by listing the set of concepts, and a set of relations with their domain and range constraints given by the ontology. For example, for the test case in the movie ontology, concepts will be a list such as a film, film genre, genre, film production company, film award, human etc. and the relations will be a list such as director(film, human), cast_member(film, human), award_received(film, award), genre(film, genre), production_company(film, film production company), etc. Throughout

[4] https://github.com/juncongmoo/pyllama.

[5] https://github.com/lm-sys/FastChat.

[6] https://github.com/tloen/alpaca-lora.

the prompt, we use the *relation(subject, object)* notation for representing relations and expect the model to follow the notation in the output.

Demonstrative Examples. This part of the prompt is used to provide the LLM with an example to show an input sentence and the expected output. LLMs are capable of In-Context Learning where they learn the task and output format from the examples provided in the prompt. The examples are taken from the training data for each of the datasets based on their similarity to the test sentence. We have used sentence similarities using Sentence Transformers (SBERT) [36] with the T5-XXL model [33]. For instance, given a test sentence such as "Super Capers, a film written by Ray Griggs, is 98 min long.", it can find the most similar sentence in training data, "English Without Tears, written by Terence Rattigan, runs 89 min." with it's aligned triples. Example output follows the same relation notation.

Test Sentence. Finally, the prompt contains the test sentence from which we want to extract the facts complying with the ontology. Similar to the example, the prompt ends with a "Test Output:"where the model is expected to generate the facts in the sentence following the same format as in the example sentence.

5.3 Evaluation Results

We run inferences for automatically generated prompts for both *Wikidata-TekGen* and *DBpedia-WebNLG* corpora and calculate the metrics discussed in Sect. 4: Precision (P), Recall (R), F1, Ontology Conformance (OC), Subject/Relation/Object Hallucinations (SH/RH/OH). Table 2 illustrates the average values across all ontologies in a given dataset. As discussed in Sect. 3, three different settings in *Wikidata-TekGen* dataset: all test cases (All), manually validated and cleaned subset (selected), and unseen sentences (Unseen) which annotated in the "Variant" column.

Each row in Table 2 is an aggregation of results from test cases across multiple ontologies (10 for Wikidata-TekGen and 19 for DBpedia-WebNLG) and Table 3 shows the results at each individual ontology level for the first row of Table 2, *i.e.*, Wikidata-TekGen - Vicuna - All. For brevity, ontology-level results for other rows are included in the project Wiki[7].

From the results, some initial observations from Table 2 on the different datasets and LLM models:

- Precision, Recall and F1 score have low intermediate values
- Ontology Conformance is pretty high in almost all entries
- Subject, Relation, Object Hallucination is relatively low

These results should be further analyzed to understand the different capbalities and limitations of LLMs in KG generation from text. An in-depth analysis of the results is out of scope of this paper due to space limitations and we expect the system papers using the benchmark to provide insights and conclusions on

[7] https://github.com/cenguix/Text2KGBench/wiki.

Table 2. This table summarizes average evaluation metrics for all ontologies in Wikidata-TekGen and the DBpedia-WebNLG datasets.

Dataset	Model	Variant[a]	Fact Extraction			OC	Hallucinations		
			P	R	F1		SH	RH	OH
Wikidata-TekGen	Vicuna	All	0.38	0.34	0.35	0.83	0.17	0.17	0.17
		Selected	0.42	0.39	0.38	0.84	0.11	0.16	0.14
		Unseen	0.32	0.32	0.32	0.86	0.07	0.14	0.14
	Alapaca LoRA	All	0.32	0.26	0.27	0.87	0.18	0.13	0.17
		Selected	0.33	0.27	0.28	0.87	0.12	0.13	0.17
		Unseen	0.22	0.22	0.22	0.86	0.09	0.14	0.26
DBpedia-WebNLG	Vicuna		0.34	0.27	0.30	0.93	0.12	0.07	0.28
	Alpaca-LoRA		0.32	0.23	0.25	0.91	0.16	0.09	0.38

[a] Refer to Sect. 3 for details.

Table 3. Results for Vicuna LLM All Test Cases. Numbers in bold identify the best results for each metric. Numbers underlined identify worst results.

Ontology	Fact Extraction			OC	Hallucinations		
	P	R	F1		SH	RH	OH
1. Movie Ontology	0.33	0.23	0.25	0.89	0.26	0.11	0.26
2. Music Ontology	0.42	0.28	0.32	0.94	0.16	**0.06**	0.22
3. Sport Ontology	0.57	0.52	0.52	0.85	0.22	0.15	0.13
4. Book Ontology	0.31	0.25	0.26	0.92	0.16	0.08	0.23
5. Military Ontology	0.24	0.25	0.24	0.8	0.19	0.2	0.26
6. Computer Ontology	0.38	0.35	0.35	0.85	0.15	0.15	0.11
7. Space Ontology	**0.68**	**0.67**	**0.66**	0.93	0.15	0.07	**0.08**
8. Politics Ontology	0.34	0.32	0.33	0.92	0.17	0.08	0.15
9. Nature Ontology	0.25	0.27	0.25	0.68	**0.1**	0.32	0.14
10 Culture Ontology	0.31	0.32	0.31	0.59	0.15	0.41	0.12
Ontologies Average	0.38	0.34	0.35	0.83	0.17	0.17	0.17

this aspect. As we have used LLM models as is without any fine-tuning, prompt tuning or semantic validation, we believe there is a large room for improvements.

5.4 Error Analysis

We have performed an initial error analysis to understand the errors made by the models and Table 4 shows some examples for different types of errors. In addition, we noticed that there are some false positives in hallucination due to LLMs expanding acronyms, for example, the sentence can have "NATO" where the model generates "North Atlantic Treaty Organization" as a subject. We plan to consider acronyms, aliases, etc. in hallucination calculations in the future.

Table 4. Examples of errors from the Vicuna13B model with Wikidata-TekGen

Sentence	Triple	Error Type
Aparajito won 11 international awards, including the Golden Lion and Critics Award at the Venice Film Festival, becoming the first ever film to win both.	award_received(Aparajito, Venice Film Festival)	An incorrect fact extracted. The model mistook the film festival for an award.
The Gallopin Gaucho was a second attempt at success by co-directors Walt Disney and Ub Iwerks.	directed(The Gallopin Gaucho,Walt Disney)	Ontology conformance error. The canonical relation is the director.
American Born Chinese is a graphic novel by Gene Luen Yang.	narrative_location(American Born Chinese, San Francisco)	Object hallucination. Neither the object nor the relation is mentioned in the text.
Schreck was a founding member of the Sturmabteilung.	member_of_political_party (Hermann Goring, Sturmabteilung)	Subject hallucination. Hermann Goring is not mentioned in the text.

6 Related Work

The primary aim of the knowledge graph generation task is to extract structured information from heterogeneous sources. This section will explore the Relation Extraction Benchmarks, Foundation Models for Knowledge Graph Generation and Semi-Automatic/Automatic Knowledge Graph Completion (KBC)-KG-triple generation. Relation extraction has made substantial use of the following datasets such as the New York Times (NYT)/NYT-FB dataset [32] [37] [30], TAC Relation Extraction Dataset (TACRED) [55], Large-Scale Document-Level Relation Extraction Dataset(DocRED) [53], The WEB-NLG dataset [13], FewRel dataset [14], FewRel 2.0 [12]. The relation extraction benchmarks that exist for the scientific domain are SciERC dataset [29] and SCIREX [20]. The SCIREX dataset is intended to detect both binary and n-ary relations between entities and concepts, while the SciERC dataset is intended to identify binary relations between entities in scientific papers. There are few datasets that cover multiple languages, such as Multilingual LAMA (Language Model Analysis) dataset [22], which cover 53 languages, MiLER SMiLER(Samsung Multi-Lingual Entity and Relation Extraction dataset [40] covers 14 languages, DiS-ReX [4] covers 4 languages. Through entity linking, Knowledge-Enhanced Relation Extraction Dataset (KERED) [27] gives knowledge context for entities and annotates each sentence with a relational fact. This dataset consists of NYT10m, Wikidata [48] (Wiki80 and Wiki20m).

Relation extraction benchmark datasets can be used to evaluate the performance of foundation models. A survey paper [52] has explored the history of these foundation models and summarizes different tasks. Foundation models are generally categorized into two categories: Encoder-only or Encoder-Decoder (BERT style) and Decoder-only (GPT style) [52]. BERT-style models are still challenging as they are under development and mostly available as open-source. They are considered as Mask Language Models that include RoBERTa [28], BERT [10], and T5 [35]. Decoder-only (GPT style) models (GPT-3 [6], PaLM [8], OPT [54] and BLOOM [39]) generally need finetuning on datasets of the particular downstream task. Brown et al. [6] have trained GPT-3 (an autoregressive language model) with 175 billion parameters and also tested its performance with the few-shot setting. Jeremy and Sebastian [18] have proposed an effective transfer learning method, Universal Language Model Fine-tuning (ULMFiT), for any NLP task. Brian et al. [25] explores the prompt tuning to learn soft prompts for adapting language models. Soft prompts are learned by back propagation, while GPT-3 uses discrete text prompts [49].

Vicuna [7] is an open-source chatbot and it is trained by fine-tuning LLaMA. It is shown by evaluation that Vicuna has performed more than 90% quality of Google Bard and OpenAI ChatGPT compared to other models like LLaMA and Alpaca. Alpaca [42] has been introduced as a strong, replicable instruction-following model. It is fine-tuned from the LLaMA 7B model on 52K instruction-following demonstrations.

KBC and KG-triple generation have become a hot research field with the synergy/integration with LLMs. The possibilities are limitless regarding the automatic generation of new triples via the use of LLMs and the only *"Achilles Heel"* consists of computer resources required for integrating both systems. In [5] is presented a system that generates automatically triples from natural language and code completion tasks. In this case, it is presented as input code excerpts denoting class and function definitions. They consider the use of neural networks present in pre-trained LLM's as *"black boxes"*. In [2] is presented a system that uses a GPT3 LLM with the aim of building a knowledge base semi-automatically via a multi-step process combining customized prompting techniques for predicting missing objects in triples where subjects and relations are given. In [23], the authors perform a qualitative study of large language models using ChatGPT for various tasks including KG population, KG completion, triple or fact verification and identify some challenges such as hallucination, fairness and bias, and high computational cost. And finally, we include reference [47] in this section where it is presented a benchmark dataset for assessing Knowledge Base Completion (KBC) potential for language models (LM).

7 Conclusion and Future Work

In this paper, we presented *Text2KGBench*, a benchmark for evaluating capabilities of LLMs for extracting facts from a text corpora guided by an ontology.

Limitations. In this version, we have only considered smaller-sized ontologies by design to cater for the token size limitations of LLMs. Nevertheless, in practice, there are quite larger ontologies in domains such as medicine. In future versions, we plan to include cases with much larger ontologies which will require systems to automatically select the portion of the ontology or the set of axioms that are relevant to the given input text. In addition, there is research on extending the capabilities of LLMs to handle longer contexts such as Unlimiformer [3]. Furthermore, in this version, we have separated the OWL/RDF representations of KGs by verbalizing ontologies and triples. In future versions, we will test LLMs on handing these representations directly without pre/post-processing.

Future Work. One important aspect when it comes to foundation models is bias and fairness. In future work, we would like to further extend our benchmark considering different bias variables such as gender, race/ethnicity, geographic location, etc. and create contrastive test cases to verify the fairness of LLMs when generating Knowledge Graphs from text. In other words, we would like to systematically evaluate if this process performs better for a certain subgroup based on their gender, demographics, or socioeconomic status. Furthermore, we will plan to measure more reasoning capabilities when performing fact extraction and KG generation. We plan to extend the benchmark with a dataset that requires more semantic reasoning to perform the task. In the Text2KGBench benchmark we have currently focused on available open-source LLM models. In addition, we plan to compare both LLM base-lines, Vicuna-13B and Alpaca-Lora-13B, and any emerging new open-source LLMs to the commercial OpenAI's ChatGPT[8] and GPT-4[9] LLMs.

Impact: With the popularity of GPT-like LLMs, there is a big enthusiasm for using such models jointly with KGs and for constructing KGs. Authors firmly believe that ontology-driven KG construction from text leveraging LLMs will be of interest to the Semantic Web community. To the best of our knowledge, *Text2KG* is the first benchmark for this task. We provide all the resources necessary for using and further extending the benchmark with improvements. Authors anticipate that this will inspire research in this direction by providing a way to measure and compare the performance of different approaches.

Reusability and Sustainability: There are two ongoing workshops related to KG generation from text, Text2KG[10] at ESWC and NLP4KGC[11] at the Web Conference. Furthermore, there is a proposed special issue[12] on this theme at Semantic Web journal. This will be a useful resource for evaluating approaches presented in those venues. As the authors are also co-organizers of these events, they plan

[8] https://openai.com/blog/chatgpt.

[9] https://openai.com/gpt-4.

[10] https://aiisc.ai/text2kg2023/.

[11] https://sites.google.com/view/nlp4kg/.

[12] https://www.semantic-web-journal.net/blog/special-issue-knowledge-graph-generation-text.

to maintain and provide improved future versions of the data in collaboration with those workshops. It's also important to note that the authors and organizations of the aforementioned workshops are not from a single organization but distributed across multiple organizations and making the proposed resource not dependent on a single organization. The code used to generate the resource is available making it possible for anyone to reproduce, improve or create derived work from it.

Resource Availability Statement: Text2KGBench dataset is available from zenodo[13], and the code that is used to generate the benchmark, evaluation scripts, baselines, LLM outputs, evaluation results are available from Github[14]. Raw datasets we used are TekGen corpus[15] and WebNLG corpus[16]. The LLM models we used are LLaMA[17] to derive Vicuna-13B[18] and Alpaca-LoRA-13B[19].For sentence similarity we used SBERT[20] with T5-XXL model[21].

References

1. Agarwal, O., Ge, H., Shakeri, S., Al-Rfou, R.: Knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 3554–3565. Association for Computational Linguistics, Online (2021). https://doi.org/10.18653/v1/2021.naacl-main.278. https://aclanthology.org/2021.naacl-main.278
2. Alivanistos, D., Santamaría, S.B., Cochez, M., Kalo, J.C., van Krieken, E., Thanapalasingam, T.: Prompting as probing: using language models for knowledge base construction. arXiv preprint arXiv:2208.11057 (2022)
3. Bertsch, A., Alon, U., Neubig, G., Gormley, M.R.: Unlimiformer: long-range transformers with unlimited length input (2023)
4. Bhartiya, A., Badola, K., et al.: Dis-rex: a multilingual dataset for distantly supervised relation extraction. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 849–863 (2022)
5. Bi, Z., et al.: Codekgc: code language model for generative knowledge graph construction. arXiv preprint arXiv:2304.09048 (2023)
6. Brown, T., et al.: Language models are few-shot learners. Adv. Neural. Inf. Process. Syst. **33**, 1877–1901 (2020)
7. Chiang, W.L., et al.: Vicuna: an open-source chatbot impressing GPT-4 with 90%* ChatGPT quality (2023). https://vicuna.lmsys.org

[13] https://zenodo.org/record/7916716#.ZFrX5ezML0r.
[14] https://github.com/cenguix/Text2KGBench.
[15] https://paperswithcode.com/dataset/tekgen.
[16] https://gitlab.com/shimorina/webnlg-dataset/-/tree/master/release_v3.0/en.
[17] https://github.com/juncongmoo/pyllama.
[18] https://github.com/lm-sys/FastChat.
[19] https://github.com/tloen/alpaca-lora.
[20] https://www.sbert.net/.
[21] https://huggingface.co/sentence-transformers/gtr-t5-xxl.

8. Chowdhery, A., et al.: Palm: scaling language modeling with pathways. arXiv preprint arXiv:2204.02311 (2022)
9. Christiano, P.F., Leike, J., Brown, T., Martic, M., Legg, S., Amodei, D.: Deep reinforcement learning from human preferences. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
10. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
11. Dimou, A., Vander Sande, M., Colpaert, P., Verborgh, R., Mannens, E., Van de Walle, R.: RML: a generic language for integrated RDF mappings of heterogeneous data. Ldow **1184** (2014)
12. Gao, T., et al.: Fewrel 2.0: towards more challenging few-shot relation classification. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 6250–6255 (2019)
13. Gardent, C., Shimorina, A., Narayan, S., Perez-Beltrachini, L.: Creating training corpora for NLG micro-planners. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 179–188. Association for Computational Linguistics (2017). https://doi.org/10.18653/v1/P17-1017. http://www.aclweb.org/anthology/P17-1017
14. Han, X., et al.: Fewrel: a large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 4803–4809 (2018)
15. Hitzler, P.: Neuro-symbolic artificial intelligence: the state of the art (2022)
16. Hogan, A., et al.: Knowledge graphs. ACM Comput. Surv. (CSUR) **54**(4), 1–37 (2021)
17. Howard, J., Ruder, S.: Universal language model fine-tuning for text classification. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, pp. 328–339 (2018)
18. Howard, J., Ruder, S.: Universal language model fine-tuning for text classification. arXiv preprint arXiv:1801.06146 (2018)
19. Hu, E.J., et al.: LoRA: low-rank adaptation of large language models. In: International Conference on Learning Representations (ICLR 2022) (2022). https://openreview.net/forum?id=nZeVKeeFYf9
20. Jain, S., van Zuylen, M., Hajishirzi, H., Beltagy, I.: Scirex: a challenge dataset for document-level information extraction. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 7506–7516 (2020)
21. Ji, Z., et al.: Survey of hallucination in natural language generation. ACM Comput. Surv. **55**(12) (2023). https://doi.org/10.1145/3571730
22. Kassner, N., Dufter, P., Schütze, H.: Multilingual lama: investigating knowledge in multilingual pretrained language models. In: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pp. 3250–3258 (2021)
23. Khorashadizadeh, H., Mihindukulasooriya, N., Tiwari, S., Groppe, J., Groppe, S.: Exploring in-context learning capabilities of foundation models for generating knowledge graphs from text. In: Proceedings of the Second International Workshop on Knowledge Graph Generation from Text, Hersonissos, Greece, pp. 132–153. CEUR (2023)
24. Lester, B., Al-Rfou, R., Constant, N.: The power of scale for parameter-efficient prompt tuning. In: Proceedings of the 2021 Conference on Empirical Methods in

Natural Language Processing, pp. 3045–3059. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic (2021). https://doi.org/10.18653/v1/2021.emnlp-main.243. https://aclanthology.org/2021.emnlp-main.243

25. Lester, B., Al-Rfou, R., Constant, N.: The power of scale for parameter-efficient prompt tuning. arXiv preprint arXiv:2104.08691 (2021)
26. Li, X.L., Liang, P.: Prefix-tuning: optimizing continuous prompts for generation. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers) abs/2101.00190 (2021)
27. Lin, Y., et al.: Knowledge graph enhanced relation extraction datasets. arXiv preprint arXiv:2210.11231 (2022)
28. Liu, Y., et al.: Roberta: a robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
29. Luan, Y., He, L., Ostendorf, M., Hajishirzi, H.: Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 3219–3232 (2018)
30. Marcheggiani, D., Titov, I.: Discrete-state variational autoencoders for joint discovery and factorization of relations. Trans. Assoc. Comput. Linguist. **4**, 231–244 (2016)
31. Min, S., et al.: Rethinking the role of demonstrations: what makes in-context learning work? In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Abu Dhabi, United Arab Emirates, pp. 11048–11064. Association for Computational Linguistics (2022). https://aclanthology.org/2022.emnlp-main.759
32. Mintz, M., Bills, S., Snow, R., Jurafsky, D.: Distant supervision for relation extraction without labeled data. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, pp. 1003–1011 (2009)
33. Ni, J., et al.: Large dual encoders are generalizable retrievers. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates (2022)
34. Ouyang, L., et al.: Training language models to follow instructions with human feedback. arXiv preprint arXiv:2203.02155 (2022)
35. Raffel, C., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res. **21**(1), 5485–5551 (2020)
36. Reimers, N., Gurevych, I.: Sentence-BERT: sentence embeddings using Siamese BERT-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics (2019). http://arxiv.org/abs/1908.10084
37. Riedel, S., Yao, L., McCallum, A.: Modeling relations and their mentions without labeled text. In: Balcázar, J.L., Bonchi, F., Gionis, A., Sebag, M. (eds.) ECML PKDD 2010. LNCS (LNAI), vol. 6323, pp. 148–163. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15939-8_10
38. Sahoo, S.S., et al.: A survey of current approaches for mapping of relational databases to RDF. W3C RDB2RDF Incubator Group Report, vol. 1, pp. 113–130 (2009)
39. Scao, T.L., et al.: Bloom: a 176B-parameter open-access multilingual language model. arXiv preprint arXiv:2211.05100 (2022)

40. Seganti, A., Firląg, K., Skowronska, H., Satława, M., Andruszkiewicz, P.: Multilingual entity and relation extraction dataset and model. In: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pp. 1946–1955 (2021)
41. Stiennon, N., et al.: Learning to summarize with human feedback. Adv. Neural. Inf. Process. Syst. **33**, 3008–3021 (2020)
42. Taori, R., et al.: Stanford alpaca: an instruction-following llama model (2023). https://github.com/tatsu-lab/stanford_alpaca
43. Tiwari, S., et al. (eds.): Proceedings of the 1st International Workshop on Knowledge Graph Generation From Text and the 1st International Workshop on Modular Knowledge co-located with 19th Extended Semantic Conference (ESWC 2022), Hersonissos, Greece, May 30th, 2022, CEUR Workshop Proceedings, vol. 3184. CEUR-WS.org (2022). http://ceur-ws.org/Vol-3184
44. Tiwari, S., et al. (eds.): Proceedings of the 2nd International Workshop on Knowledge Graph Generation From Text and the International BiKE Challenge co-located with 20th Extended Semantic Conference (ESWC 2023), Hersonissos, Greece, May 28th, 2023, CEUR Workshop Proceedings, vol. 3447. CEUR-WS.org (2023). http://ceur-ws.org/Vol-3447
45. Touvron, H., et al.: Llama: open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023)
46. Vakaj, E., Tiwari, S., Mihindukulasooriya, N., Ortiz-Rodríguez, F., Mcgranaghan, R.: NLP4KGC: natural language processing for knowledge graph construction. In: Companion Proceedings of the ACM Web Conference 2023, pp. 1111–1111 (2023)
47. Veseli, B., Singhania, S., Razniewski, S., Weikum, G.: Evaluating language models for knowledge base completion. arXiv preprint arXiv:2303.11082 (2023)
48. Vrandečić, D., Krötzsch, M.: Wikidata: a free collaborative knowledgebase. Commun. ACM **57**(10), 78–85 (2014)
49. Wang, A., et al.: Superglue: a stickier benchmark for general-purpose language understanding systems. In: Advances in Neural Information Processing Systems, vol. 32 (2019)
50. Wang, Y., et al.: Self-instruct: aligning language model with self generated instructions. arXiv preprint arXiv:2212.10560 (2022)
51. Xie, S.M., Raghunathan, A., Liang, P., Ma, T.: An explanation of in-context learning as implicit Bayesian inference. arXiv preprint arXiv:2111.02080 (2021)
52. Yang, J., et al.: Harnessing the power of LLMS in practice: a survey on ChatGPT and beyond. arXiv preprint arXiv:2304.13712 (2023)
53. Yao, Y., et al.: Docred: a large-scale document-level relation extraction dataset. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 764–777 (2019)
54. Zhang, S., et al.: OPT: open pre-trained transformer language models. arXiv preprint arXiv:2205.01068 (2022)
55. Zhang, Y., Zhong, V., Chen, D., Angeli, G., Manning, C.D.: Position-aware attention and supervised data improve slot filling. In: Conference on Empirical Methods in Natural Language Processing (2017)
56. Zheng, L., et al.: Judging LLM-as-a-judge with MT-bench and chatbot arena. arXiv preprint arXiv:2306.05685 (2023)