Enhancing Knowledge Graph Construction: Evaluating with Emphasis on Hallucination, Omission, and Graph Similarity Metrics

Hussam Ghanem, Christophe Cruz

Enhancing Knowledge Graph Construction: Evaluating with Emphasis on Hallucination, Omission, and Graph Similarity Metrics

Hussam Ghanem[1] and Christophe Cruz[1]

ICB, UMR 6306, CNRS, Université de Bourgogne, 21000 Dijon, France
https://icb.u-bourgogne.fr/

Abstract. Recent advancements in large language models have demonstrated significant potential in the automated construction of knowledge graphs from unstructured text. This paper builds upon our previous work [16], which evaluated various models using metrics like precision, recall, F1 score, triple matching, and graph matching, and introduces a refined approach to address the critical issues of hallucination and omission. We propose an enhanced evaluation framework incorporating BERTScore for graph similarity, setting a practical threshold of 95% for graph matching. Our experiments focus on the Mistral model, comparing its original and fine-tuned versions in zero-shot and few-shot settings. We further extend our experiments using examples from the KELM-sub training dataset, illustrating that the fine-tuned model significantly improves knowledge graph construction accuracy while reducing the exact hallucination and omission. However, our findings also reveal that the fine-tuned models perform worse in generalization tasks on the KELM-sub dataset. This study underscores the importance of comprehensive evaluation metrics in advancing the state-of-the-art in knowledge graph construction from textual data.

Keywords: Text-to-Knowledge Graph, Large Language Models, Zero-Shot Prompting, Few-Shot Prompting, Fine-Tuning, Hallucination

1 Introduction

Knowledge Graphs (KGs) play a crucial role in organizing complex information across diverse domains, such as question answering, recommendations, semantic search, etc. However, the ongoing challenge persists in constructing them, particularly as the primary sources of knowledge are embedded in unstructured textual data such as press articles, emails, and scientific journals. This challenge can be addressed by adopting an information extraction approach, sometimes implemented as a pipeline. It involves taking textual inputs, processing them using Natural Language Processing (NLP) techniques, and leveraging the acquired knowledge to construct or enhance the KG.

In-context learning, as discussed by [7], coupled with prompt design, involves telling a model to execute a new task by presenting it with only a few demon-

strations of input-output pairs during inference. Instruction fine-tuning methods, exemplified by InstructGPT [8] and Reinforcement Learning from Human Feedback (RLHF) [9], markedly enhance the model's ability to comprehend and follow a diverse range of written instructions. Numerous large language models (LLMs) have been introduced in the last year, as highlighted by [3], particularly within the ChatGPT [10] like models, which includes GPT-3 [11], LLaMA [12], Mistral [15], and Starling [17]. These models can be readily repurposed for KG construction from text by employing a prompt design that incorporates instructions and contextual information.

The task of converting textual information into structured KGs has gained significant traction with the advent of LLMs. These models offer unprecedented capabilities in understanding and generating human-like text, making them invaluable for a variety of NLP applications. Our previous work [16] explored different approaches to the Text-to-Knowledge Graph (T2KG) construction task, including Zero-Shot Prompting (ZSP) [19], Few-Shot Prompting (FSP) [6], and Fine-Tuning (FT) [4] of LLMs, employing models such as Llama2 [12], Mistral [15], and Starling [17]. In this work, we will include a little state of the art on contributions that use these three approaches (Section 2).

While traditional metrics like precision, recall, F1 score, triple matching, and graph matching provide a baseline for evaluating these models, they often overlook critical qualitative aspects of the generated graphs, such as hallucinations (incorrect or spurious triples) and omissions (missing relevant triples). Addressing these gaps, our current study introduces a refined evaluation framework that incorporates refined hallucination and omission metrics, and also incorporates BERTScore to measure the similarity between generated and ground truth graphs, setting an 95% similarity threshold for graph matching. This nuanced approach aims to provide a more comprehensive assessment of the models' performance in generating accurate and complete knowledge graphs.

In this paper, we specifically focus on comparing the original Mistral model and our finetuned Mistral (from our previous work) under zero-shot and few-shot settings. Additionally, we extend our experiments to include the KELM-sub dataset, utilizing few-shot examples to demonstrate that fine-tuning on a specific domain (WebNLG) significantly enhances performance when applied to related but distinct datasets with just few examples.

The present study is organized as follows, Section 2 presents a comprehensive overview of the current state-of-the-art approaches for Text to KG (T2KG) Construction and its evaluation metrics. In the Section 3, we present the general architecture of our proposed implementation (method), with datasets, metrics, and experiments. Section 4 then encapsulates the findings and discussions, presenting the culmination of results. Finally, Section 5 critically examines the strengths and limitations of these techniques.

2 Background

The current state of research on knowledge graph construction using LLMs is discussed. Three main approaches are identified: Zero-Shot, Few-Shot, and Fine-Tuning. Each approach has its own challenges, such as maintaining accuracy without specific training data or ensuring the robustness of models in diverse real-world scenarios. Evaluation metrics used to assess the quality of constructed KGs are also discussed, including semantic consistency and linguistic coherence. This section highlight methods and metrics to construct KGs and evaluate the result.

2.1 Zero Shot

Zero Shot methods enable KG construction without task-specific training data, leveraging the inherent capabilities of LLMs. [19] introduce an innovative approach using LLMs for knowledge graph construction, employing iterative zero-shot prompting for scalable and flexible KG construction. [20] evaluate the performance of LLMs, specifically GPT-4 and ChatGPT, in KG construction and reasoning tasks, introducing the Virtual Knowledge Extraction task and the VINE dataset, but they do not take into account open sourced LLMs as LLaMA [12]. [24] address the limitations of existing generative knowledge graph construction methods by leveraging large generative language models trained on structured data. The most of these approaches having the same limitation, which is the use of closed and huge LLMs as ChatGPT or GPT4 for this task. Challenges in this area include maintaining accuracy without specific training data and addressing nuanced relationships between entities in untrained domains.

2.2 Few Shot

Few Shot methods focus on constructing KGs with limited training examples, aiming to achieve accurate knowledge representation with minimal data. [6] introduce PiVe, a framework enhancing the graph-based generative capabilities of LLMs, and the authors create a verifier which is responsable to verifie the results of LLMs with multi-iteration type. [29] investigate LLMs' application in relation labeling for e-commerce Knowledge Graphs (KGs). As ZSP approaches, FSP approaches use closed and huge LLMs as ChatGPT or GPT4 [10] for this task. Challenges in this area include achieving high accuracy with minimal training data and ensuring the robustness of models in diverse real-world scenarios.

2.3 Fine-Tuning

Fine-Tuning methods involve adapting pre-trained language models to specific knowledge domains, enhancing their capabilities for constructing KGs tailored to particular contexts. [4] present a case study automating KG construction for compliance using BERT-based models. This study emphasizes the importance

of machine learning models in interpreting rules for compliance automation. [31] propose Knowledge Graph-Enhanced Large Language Models (KGLLMs), enhancing LLMs with KGs for improved factual reasoning capabilities. These approaches that applied FT, they do not use new generations of LLMs, specially, decoder only LLMs as Llama, and Mistral. Challenges in this domain include ensuring the scalability, interpretability, and robustness of fine-tuned models across diverse knowledge domains.

2.4 Evaluation metrics

As we employ LLMs to construct KGs, and given that LLMs function as Natural Language Generation (NLG) models, it becomes imperative to discuss NLG criteria. In NLG, two criteria [32] are used to assess the quality of the produced answers (triples in our context).

The first criterion is semantic consistency or Semantic Fidelity, which includes:

- **Hallucination**: Presence of information (facts) in the generated text that is absent in the input data.
- **Omission**: Omission of information present in the input data from the generated text.
- **Redundancy**: Repetition of information in the generated text (not considered in our evaluation).
- **Accuracy**: Exact match between the input and generated text without modification.
- **Ordering**: Sequence of information in the generated text differing from the input data (not considered in our evaluation).

The second criterion is linguistic coherence or Output Fluency, which evaluates the fluidity and linguistic correctness of the generated text. This criterion is not considered in our evaluation.

In their experiments, [3] calculated three hallucination metrics - subject hallucination, relation hallucination, and object hallucination - using preprocessing steps like stemming. They used the ground truth ontology and test sentence to determine if an entity or relation is present, considering any disparity between them as hallucination.

The authors of [6] evaluated their experiments using several metrics, including Triple Match F1 (T-F1), Graph Match F1 (G-F1), G-BERTScore (G-BS) from [33], and Graph Edit Distance (GED) from [35]. The GED metric measures the distance between the predicted and ground-truth graphs by calculating the number of edit operations needed to transform one into the other. To adhere to the semantic consistency criterion, we use the terms "omission" and "hallucination" instead of "addition" and "deletion," respectively.

3 Propositions

This section outlines our approach to evaluate the quality of generated KGs using metrics like T-F1, G-F1, G-BS, and GED. We also discuss the use of Op-

timal Edit Paths (OEP) to determine the precise number of operations needed to transform the predicted graph into an identical representation of the ground-truth graph. This method helps in calculating omissions and hallucinations in the generated graphs. Unlike our previous work where we marked a single hallucination or omission per generated graph, we now calculate the exact number of hallucinations and omissions for each generated graph (Fig. 1. Previously, we used examples from the WebNLG+2020 dataset [38] for testing with FSP techniques and trained LLMs using the FT technique. In this work, we take the best fine-tuned model (Mistral) from our previous work and apply zero/few-shot learning, comparing it with the original Mistral. Examples for few-shot learning are taken from WebNLG+2020 and the KELM-sub training dataset, and inference is applied on both datasets. We then compare these results with our previous work where models were applied on WebNLG+2020 and KELM-sub using examples from the WebNLG+2020 training dataset.

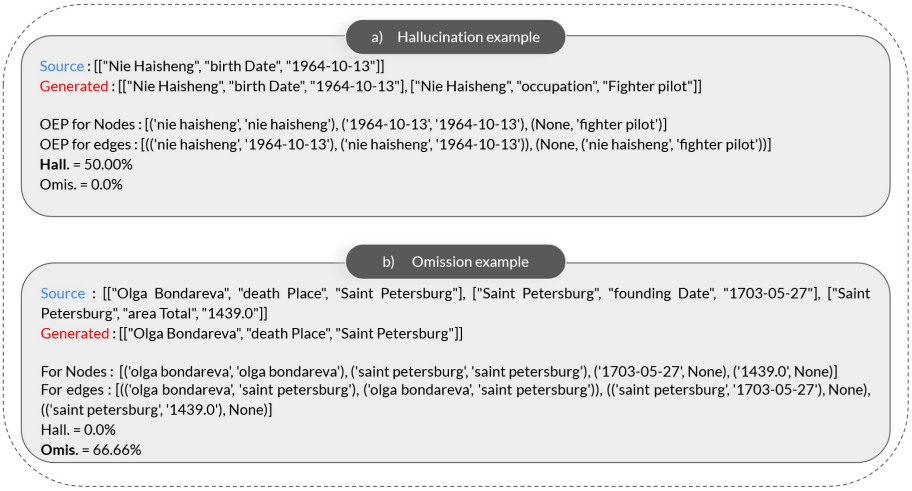

Fig. 1. Results examples

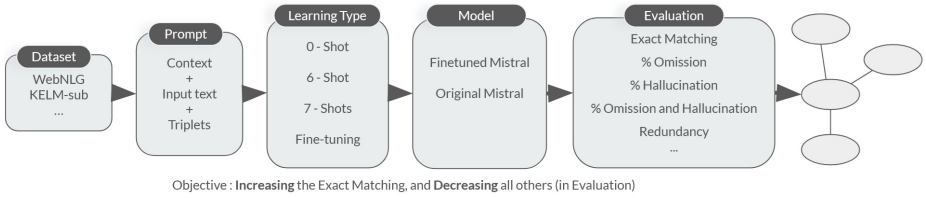

Fig. 2. Overall experimentation's process

3.1 Overall experimentation's process

In our previous work, we leveraged the WebNLG+2020 and KELM-sub datasets, specifically the version curated by [6]. Their preparation of graphs in lists of triples proves beneficial for evaluation purposes. We utilize these lists and employ NetworkX [39] to transform them back into graphs, facilitating evaluations on the resultant graphs. This step is instrumental in performing ZSP, FSP, and FT LLMs on these datasets. In this work, we will use examples from the training dataset of KELM-sub to do few-shot learning on the original and the finetuned (from our previous work) Mistral model.

Fig. 2 illustrates the different stages of our experimentation process, including data preparation, model selection, training, validation, and evaluation. The process begins with data preparation, where the WEBNLG dataset is preprocessed and split into training, validation, and test sets. Next, the learning type is selected, and different models are trained using the training set. The trained models are then evaluated on the validation set to evaluate their performance. Finally, the best-performing model is selected and validated on the test set to estimate its generalization ability.

3.2 Prompting learning

In this phase, we use ZSP and FSP techniques on LLMs to evaluate their proficiency in extracting triples for KG construction. We merge examples from the KELM-sub test dataset with our adapted prompt, strategically modified for contextual guidance without a support ontology description, as demonstrated by [3]. The prompts for ZSP and FSP are shown in Fig. 3(a) and Fig. 3(b).

For ZSP, we started with the method from [6], using the directive "Transform the text into a semantic graph" and enhanced it with additional sentences for our LLMs (Fig. 3(a)). For FSP, we used 6-shot learning, corresponding to the maximum KG size in KELM-sub, feeding the prompt with six examples of varying sizes (Fig. 3(b)).

3.3 Postprocessing

To evaluate the generated KGs against ground-truth KGs, we clean the LLM outputs by transforming generated graphs into organized lists of triples and transferring them to textual documents. This rule-based processing removes corrupted text outside the lists of triples, optimizing our evaluation process for metrics like G-F1, GED, and OEP (Section 3.4).

In our previous work, instructing LLMs to produce lists of triples sometimes resulted in unstructured text, which we addressed by substituting the generated text with an empty list of triples ('[["","",""]]'). This approach, however, underestimated hallucinations. In the current work, as illustrated in Fig. 1, we calculate the exact hallucination and omission for each generated graph through qualitative evaluation of two randomly generated graphs.

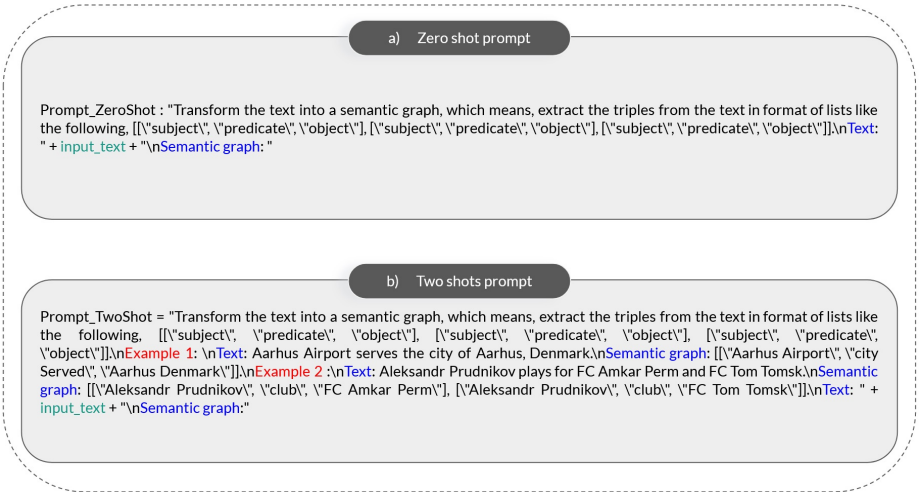

Fig. 3. Prompting examples

3.4 Experiment's evaluation

To evaluate the generated graphs against ground-truth graphs, we use metrics such as T-F1, G-F1, G-BS [33], and GED [35] as in [6]. We also use Optimal Edit Paths (OEP) to calculate omissions and hallucinations in the generated graphs.

Our evaluation follows [6]'s methodology, especially in computing GED and G-F1, and involves constructing directed graphs from lists of triples using NetworkX [39]. Unlike [3], we do not use the ground truth test sentence of an ontology. Instead, we assess omissions and hallucinations using OEP, which provides the precise path of the edit, allowing exact quantification of these errors.

For example, Fig. 1 shows 2 omissions ('b)') and 1 hallucination 'a)' in using one of two paths "OEP for nodes" or OEP for edges". Previously, we incremented the global hallucination metric for all graphs if ≥ 1 hallucinations or omissions were found. In the current work, we use OEP to detect the exact percentage of hallucination or omission in a generated graph, experimenting on 2 random examples from the WebNLG+2020 test dataset (Fig. 1).

Different from our previous work, our experiments are evaluated using examples from the KELM-sub test dataset (Table 2 and Table 1). Our primary goal is to improve G-F1, T-F1, G-BS and GM-GBS metrics, while reducing GED, hallucination, and omission.

3.5 Mathematical representation of the used metrics

This study refines the metrics used for evaluating hallucinations and omissions in generated graphs and introduces a new metric, Graph Matching using Graph BERTScore (GM-GBS). In our previous work, we detailed the mathematical representation of all metrics used.

The G-BS metric evaluates graph matching by treating edges as sentences and using BERTScore to measure alignment between predicted and ground-truth edges. The F1 score for G-BS is calculated as follows:

$$R_{\text{BERT}} = \frac{1}{|x|} \sum_{x_i \in x} \max_{\hat{x}_j \in \hat{x}} x_i^T \hat{x}_j,$$

$$P_{\text{BERT}} = \frac{1}{|\hat{x}|} \sum_{\hat{x}_j \in \hat{x}} \max_{x_i \in x} x_i^T \hat{x}_j,$$

$$F1_{\text{BERT}} = \frac{2 \cdot P_{\text{BERT}} \cdot R_{\text{BERT}}}{P_{\text{BERT}} + R_{\text{BERT}}}.$$

Where R_{BERT} is the recall, and P_{BERT} is the precision.

In this work, we use G-BS to compare generated graphs with ground-truth graphs, defining graph matching with a similarity threshold of 95% to introduce GM-GBS. This approach acknowledges that entities or relations in the generated graph may be synonymous with those in the ground truth graph. Results shown in Fig.4 illustrate that even with 95% BERTScore similarity, the generated graph is nearly identical to the ground truth.

Fig. 4. Examples of the calculated GM-GBS

To calculate GM-GBS, we follow these steps: Given an array of F1 scores of G-BS $f_1, f_2, \ldots, f_n$ in `f1s_BS`, the fraction of F1 scores greater than 0.95 is calculated as follows:

1. Let $ToGrs$ be the total number of generated graphs.
2. Let f_m be the count of F1 scores that are greater than 0.95:

$$\boldsymbol{f_m = \sum_{i=1}^{N} 1(f_i > 0.95)}$$

where $1(\cdot)$ is the indicator function, which is 1 if the condition inside is true and 0 otherwise.

3. The fraction of F1 scores greater than 0.95 is given by: GM-GBS = $\frac{f_m}{N}$

For hallucinations and omissions, we use Optimal Edit Paths (OEP) to determine exact counts:

Hallucination: An edit operation is a hallucination if it adds an entity or relation not present in the gold graph. We previously used an overall hallucination metric $\boldsymbol{Hall.} = \frac{\boldsymbol{hall}}{\boldsymbol{ToGrs}}$, where *hall* is the number of graphs with hallucinations.

Omission: An edit operation is an omission if it deletes an entity or relation present in the gold graph. In the previous work the omission was computed by $\boldsymbol{Omis.} = \frac{\boldsymbol{omiss}}{\boldsymbol{ToGrs}}$, where *omiss* is the number of graphs with omissions.

In this work, we calculate exact percentages of hallucination and omission through qualitative evaluation.
Given a list of tuples lst = $[(g_1, p_1), (g_2, p_2), \ldots, (g_n, p_n)]$, where g_i represents a gold edge and p_i represents a predicted edge:

1. Let h be the number of hallucinations, where a hallucination is defined as g_i = None:

$$h = \sum_{i=1}^{n} 1(g_i = \text{None})$$

2. The exact hallucination rate is then calculated as: $\textbf{Hall\_Rate} = \frac{\boldsymbol{h}}{\boldsymbol{n}}$

Where n is the total number of edges, and $1(\cdot)$ is the indicator function, which is 1 if the condition inside is true and 0 otherwise (Same for Omis_rate).

To calculate the exact omission rate:

1. Let o be the number of omissions, where an omission is defined as p_i = None:

$$o = \sum_{i=1}^{n} 1(p_i = \text{None})$$

2. The exact omission rate is then calculated as: $\textbf{Omis\_Rate} = \frac{\boldsymbol{o}}{\boldsymbol{n}}$

4 Experiments

This section outlines the LLMs used in our experiments for ZSP and FSP and presents the experimental results.

We utilized the Mistral model from HuggingFace platform[1], specifically focusing on the finetuned Mistral model which showed the best results in our previous work. We also compared the finetuned model with the original Mistral model.

- Original Mistral-7B-v0.1: A pretrained generative text model with 7 billion parameters introduced by [15], which outperforms Llama 2 13B in various benchmarks.
- Fine-tuned Mistral-7B-v0.1: Based on the original Mistral and fine-tuned on the WebNLG+2020 training dataset, this model outperformed other fine-tuned models like Llama2 (7b and 13b) and Starling in our previous work [16].

Our evaluation also considers hallucination and omission through a linguistic lens, unlike most studies which focus on precision, recall, F1 score, triple matching, or graph matching, except for [3] which includes hallucination evaluation.

Table 1 shows that the fine-tuned Mistral performs better in both ZSP and FSP compared to the original Mistral for the T2KG construction task. The performance improves with more examples (more shots), with both finetuned and original Mistral models. Seeing the fine-tuned Mistral, it has the best performance when given 7 shots, surpassing the original Mistral by a significant margin.

As mentioned in our previous work, to corroborate these findings, in this version of our study, we assess our fine-tuned models using KELM-sub dataset for few-shot. We see that even when we gave Mistral examples from KELM-sub, it works better than zero-shot for the test dataset of WebNLG.

As depicted in Fig. 2, Hall. represents Hallucinations, while Omis. denotes Omissions.

Table 1. Comparison of performance metrics and models on WebNLG test dataset. Lower values indicate better performance for GED, Hall., and Omis.

Model — Metric	G-F1	T-F1	G-BS	GED	Hall.	Omis.	GM-GBS
Mistral-0	2.30	3.27	77.87	15.84	20.35	31.31	33.27
Mistral-7	18.72	28.44	87.54	10.13	17.88	21.14	51.88
Mistral-FT-0	31.93	44.08	86.89	8.25	**13.55**	18.27	54.97
Mistral-FT-7	**34.68**	**49.11**	**91.99**	**6.69**	14.90	14.39	57.72
Mistral-6 (KELM-sub)	7.59	12.45	81.23	16.29	61.16	**7.64**	26.86
Mistral-FT-6 (KELM-sub)	31.37	47.49	91.27	7.51	27.37	8.26	**58.40**

The G-BS consistently remains high, indicating that LLMs frequently generate text with words (entities or relations) very similar to those in the ground

[1] Hugging Face: https://huggingface.co/

truth graphs, which was one reason to use it for the GM-GBS metric. The finetuned Mistral with 7 shots achieves the highest G-F1, accurately generating approximately 35% of graphs identical to the ground truth. This model performs exceptionally well across various metrics, particularly in T-F1. Additionally, the finetuned Mistral with 6 examples from KELM-sub outperforms the finetuned Mistral with 7 examples from WebNLG+2020 using the GM-GBS metric.

In Table 2, we present the evaluation results of the original Mistral with 7-shot learning (using examples from WebNLG+2020) and the fine-tuned Mistral with zero-shot (Mistral-FT-0) and 7-shot (Mistral-FT-7) learning (also using examples from WebNLG+2020) on the KELM-sub test dataset, prepared by [6] and based on [40]. It is important to note that the experiments utilized the same prompts as previously described. The 7-shots experiments used examples from the WebNLG+2020 training dataset. These experiments aim to assess the generalization ability of the original LLMs with 7-shot learning and the fine-tuned LLMs with zero-shot and 7-shot learning across diverse domains in the T2KG construction task.

Another experiment was conducted using 6 random examples from the KELM-sub training dataset. We applied this prompt to both the original Mistral (Mistral-6) and our finetuned Mistral (Mistral-FT-6) models. As expected, Mistral-6 outperformed Mistral-7 because the examples were from the KELM-sub training dataset used in Mistral-6. However, it was interesting to observe that Mistral-FT-6 performed less effectively than Mistral-6 with the same examples. This suggests that finetuning on WebNLG domains reduces the generalizability of the LLMs.

The results in Table 2 indicate that the fine-tuned Mistral models perform less effectively than the original Mistral with 7 shots from WebNLG+2020 and with 6 shots from KELM-sub. Additionally, all fine-tuned versions of Mistral (Mistral-FT-7, Mistral-FT-0, and Mistral-FT-6) show inferior results on KELM-sub compared to WebNLG+2020. This disparity can be attributed to the presence of different relation types, with some types expressed differently in KELM-sub. To address this, we utilize G-BS to calculate the similarity between two graphs and consider them as synonyms if they are sufficiently similar (>95% of similarity). This metric, called GM-GBS (Graph Matching using Graph BERTScore), is the last metric presented in Table 2. GM-GBS indicates a higher value of graph matching. To assess the reliability of this metric, we conducted a qualitative evaluation as illustrated in Fig. 4.

Overall, unlike our previous work where we used examples from WebNLG with the original and fine-tuned models for few-shot learning, using examples from KELM-sub here shows that the results are relatively similar. This indicates that fine-tuning negatively affects the generalization capability of the models.

Qualitative results : As illustrated in Figure1, our metric precisely calculates the percentage of hallucinations and omissions in the generated graphs at the triple level. For example, if a generated graph contains 2 triples and 1 of them are not present in the ground truth graph, the hallucination rate is approximately 50%.

Table 2. Results on KELM-sub. Lower values indicate better performance for GED, Hall., and Omis.

Model — Metric	G-F1	T-F1	G-BS	GED	Hall.	Omis.	GM-GBS
Mistral-7	5.50	11.35	81.77	13.74	6.72	61.09	28.66
Mistral-FT-0	2.17	8.55	78.29	14.35	7.22	56.28	12.88
Mistral-FT-7	2.89	9.92	78.42	13.63	**6.22**	61.00	13.66
Mistral-6 (KELM-sub)	**12.00**	**31.08**	**85.49**	**10.82**	25.50	**32.44**	**38.88**
Mistral-FT-6 (KELM-sub)	4.00	17.66	84.30	12.50	11.06	48.17	36.22

Similarly, for omissions, if the generated graph is missing some triples present in the ground truth graph, the omission rate is calculated accordingly.

As previously mentioned, we use G-BS to calculate the similarity between generated and ground truth graphs. If the similarity value exceeds 95%, we consider it an exact match, based on the notion that entities or relations in the generated graph are very close to those in the ground truth graph, or what we refer to as synonyms. In Fig. 4, we present examples with varying levels of similarity, including one with approximately 95% similarity, to demonstrate that even with 95% similarity, the two graphs convey the same or very similar meanings.

5 Conclusion and perspectives

In this study, we evaluated the performance of both the original and fine-tuned Mistral models for Text-to-Knowledge Graph (T2KG) construction tasks using Zero-Shot Prompting (ZSP) and Few-Shot Prompting (FSP). Our analysis incorporated a comprehensive set of metrics, including G-F1, T-F1, G-BS, GED, along with measures for hallucinations and omissions.

Our results demonstrate that the fine-tuned Mistral model generally outperforms the original Mistral, particularly in Few-Shot scenarios. The fine-tuned Mistral with seven shots achieved superior performance across most metrics, notably improving G-F1 and T-F1 scores, which indicates a higher fidelity in generating ground truth graphs, and reflects its improved ability to produce coherent and contextually relevant outputs.

Despite these improvements, we observed that fine-tuning on domain-specific data, such as WebNLG, can negatively impact the model's generalization capabilities. This was evident from the comparative performance of the fine-tuned models on the KELM-sub dataset, where the original Mistral model with 7 shots from WebNLG+2020 outperformed the fine-tuned variants. This finding highlights the importance of balancing domain-specific fine-tuning with maintaining broad generalization.

The inclusion of the GM-GBS metric provided valuable insights into the semantic similarity between generated and ground truth graphs. Our qualitative

analysis of hallucinations and omissions further enhanced our understanding of model performance at the triple level.

Looking ahead, there are several promising avenues for further research. Refining evaluation metrics to account for synonyms of entities or relations in generated graphs could improve assessment accuracy. Additionally, leveraging LLMs for data augmentation in T2KG construction shows potential, as our experiments suggest that LLMs can maintain consistency in generating results and propose relevant triples.

Expanding evaluations to a broader range of domains and datasets can provide deeper insights into how various types of data influence model behavior and performance. Combining automated metrics with human evaluation could also offer a richer understanding of model quality, with domain experts providing valuable assessments of the relevance and accuracy of generated graphs. Exploring these directions will contribute to advancing the field of T2KG construction and enhancing the capabilities of language models in producing accurate and contextually appropriate knowledge graphs.

6 Acknowledgments

The authors thank the French company DAVI (Davi The Humanizers, Puteaux, France) for their support, and the French government for the plan France Relance funding.

References

1. A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. D. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, et al., Knowledge graphs, ACM Computing Surveys (Csur) 54 (2021) 1–37.
2. N. Noy, Y. Gao, A. Jain, A. Narayanan, A. Patterson, J. Taylor, Industry-scale knowledge graphs: Lessons and challenges: Five diverse technology companies show how it's done, Queue 17 (2019) 48–75.
3. N. Mihindukulasooriya, S. Tiwari, C. F. Enguix, K. Lata, Text2kgbench: A benchmark for ontology-driven knowledge graph generation from text, in: International Semantic Web Conference, Springer, 2023, pp. 247–265.
4. V. Ershov, A case study for compliance as code with graphs and language models: Public release of the regulatory knowledge graph, arXiv preprint arXiv:2302.01842 (2023).
5. J. H. Caufield, H. Hegde, V. Emonet, N. L. Harris, M. P. Joachimiak, N. Matentzoglu, H. Kim, S. A. Moxon, J. T. Reese, M. A. Haendel, et al., Structured prompt interrogation andrecursive extraction of semantics (spires): A method for populating knowledge bases using zero-shot learning, arXiv preprint arXiv:2304.02711 (2023).
6. J. Han, N. Collier, W. Buntine, E. Shareghi, Pive: Prompting with iterative verification improving graph-based generative capability of llms, arXiv preprint arXiv:2305.12392 (2023).
7. S. Min, X. Lyu, A. Holtzman, M. Artetxe, M. Lewis, H. Hajishirzi, L. Zettlemoyer, Rethinking the role of demonstrations: What makes in-context learning work?, arXiv preprint arXiv:2202.12837 (2022).

8. L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al., Training language models to follow instructions with human feedback, Advances in neural information processing systems 35 (2022) 27730–27744.
9. N. Stiennon, L. Ouyang, J. Wu, D. Ziegler, R. Lowe, C. Voss, A. Radford, D. Amodei, P. F. Christiano, Learning to summarize with human feedback, Advances in Neural Information Processing Systems 33 (2020) 3008–3021.
10. R. OpenAI, Gpt-4 technical report. arxiv 2303.08774, View in Article 2 (2023).
11. T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, Advances in neural information processing systems 33 (2020) 1877–1901.
12. H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al., Llama: Open and efficient foundation language models, arXiv preprint arXiv:2302.13971 (2023).
13. B. Workshop, T. L. Scao, A. Fan, C. Akiki, E. Pavlick, S. Ilić, D. Hesslow, R. Castagné, A. S. Luccioni, F. Yvon, et al., Bloom: A 176b-parameter open-access multilingual language model, arXiv preprint arXiv:2211.05100 (2022).
14. A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, et al., Palm: Scaling language modeling with pathways, Journal of Machine Learning Research 24 (2023) 1–113.
15. A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, et al., Mistral 7b, arXiv preprint arXiv:2310.06825 (2023).
16. Hussam Ghanem, Christophe Cruz, Fine-Tuning vs. Prompting: Evaluating the Knowledge Graph Construction with LLMs, International Workshop On Knowledge Graph Generation From Text (TEXT2KG), Co-located with the Extended Semantic Web Conference (ESWC), 2024.
17. B. Zhu, E. Frick, T. Wu, H. Zhu, J. Jiao, Starling-7b: Improving llm helpfulness & harmlessness with rlaif, 2023.
18. L. Tunstall, E. Beeching, N. Lambert, N. Rajani, K. Rasul, Y. Belkada, S. Huang, L. von Werra, C. Fourrier, N. Habib, et al., Zephyr: Direct distillation of lm alignment, arXiv preprint arXiv:2310.16944 (2023).
19. S. Carta, A. Giuliani, L. Piano, A. S. Podda, L. Pompianu, S. G. Tiddia, Iterative zero-shot llm prompting for knowledge graph construction, arXiv preprint arXiv:2307.01128 (2023).
20. 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).
21. B. Li, G. Fang, Y. Yang, Q. Wang, W. Ye, W. Zhao, S. Zhang, Evaluating chatgpt's information extraction capabilities: An assessment of performance, explainability, calibration, and faithfulness, arXiv preprint arXiv:2304.11633 (2023).
22. X. Wei, X. Cui, N. Cheng, X. Wang, X. Zhang, S. Huang, P. Xie, J. Xu, Y. Chen, M. Zhang,et al., Zero-shot information extraction via chatting with chatgpt, arXiv preprint arXiv:2302.10205 (2023).
23. L. Jarnac, M. Couceiro, P. Monnin, Relevant entity selection: Knowledge graph bootstrapping via zero-shot analogical pruning, in: Proceedings of the 32nd ACM InternationalConference on Information and Knowledge Management, 2023, pp. 934–944.
24. Z. Bi, J. Chen, Y. Jiang, F. Xiong, W. Guo, H. Chen, N. Zhang, Codekgc: Code language model for generative knowledge graph construction, ACM Transactions on Asian and Low-Resource Language Information Processing 23 (2024) 1–16.

25. L. Yao, J. Peng, C. Mao, Y. Luo, Exploring large language models for knowledge graph completion, arXiv preprint arXiv:2308.13916 (2023).
26. H. Khorashadizadeh, N. Mihindukulasooriya, S. Tiwari, J. Groppe, S. Groppe, Exploring in-context learning capabilities of foundation models for generating knowledge graphs from text, arXiv preprint arXiv:2305.08804 (2023).
27. S. Deng, C. Wang, Z. Li, N. Zhang, Z. Dai, H. Chen, F. Xiong, M. Yan, Q. Chen, M. Chen, et al., Construction and applications of billion-scale pre-trained multimodal business knowledge graph, in: 2023 IEEE 39th International Conference on Data Engineering (ICDE), IEEE, 2023, pp. 2988–3002.
28. M. Trajanoska, R. Stojanov, D. Trajanov, Enhancing knowledge graph construction using large language models, arXiv preprint arXiv:2305.04676 (2023).
29. J. Chen, L. Ma, X. Li, N. Thakurdesai, J. Xu, J. H. Cho, K. Nag, E. Korpeoglu, S. Kumar, K. Achan, Knowledge graph completion models are few-shot learners: An empirical study of relation labeling in e-commerce with llms, arXiv preprint arXiv:2305.09858 (2023).
30. A. Harnoune, M. Rhanoui, M. Mikram, S. Yousfi, Z. Elkaimbillah, B. El Asri, Bert based clinical knowledge extraction for biomedical knowledge graph construction and analysis, Computer Methods and Programs in Biomedicine Update 1 (2021) 100042.
31. L. Yang, H. Chen, Z. Li, X. Ding, X. Wu, Chatgpt is not enough: Enhancing large language models with knowledge graphs for fact-aware language modeling, arXiv preprint arXiv:2306.11489 (2023).
32. T. C. Ferreira, C. van der Lee, E. Van Miltenburg, E. Krahmer, Neural data-to-text generation: A comparison between pipeline and end-to-end architectures, arXiv preprint arXiv:1908.09022 (2019).
33. S. Saha, P. Yadav, L. Bauer, M. Bansal, Explagraphs: An explanation graph generation task for structured commonsense reasoning, arXiv preprint arXiv:2104.07644 (2021).
34. T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, Y. Artzi, Bertscore: Evaluating text generation with bert, arXiv preprint arXiv:1904.09675 (2019).
35. Z. Abu-Aisheh, R. Raveaux, J.-Y. Ramel, P. Martineau, An exact graph edit distance algorithm for solving pattern recognition problems, in: 4th International Conference on Pattern Recognition Applications and Methods 2015, 2015.
36. K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, Bleu: a method for automatic evaluation of machine translation, in: Proceedings of the 40th annual meeting of the Association for Computational Linguistics, 2002, pp. 311–318.
37. C.-Y. Lin, Rouge: A package for automatic evaluation of summaries, in: Text summarization branches out, 2004, pp. 74–81.
38. C. Gardent, A. Shimorina, S. Narayan, L. Perez-Beltrachini, The webnlg challenge: Gener-ating text from rdf data, in: Proceedings of the 10th international conference on natural language generation, 2017, pp. 124–133.
39. A. Hagberg, P. Swart, D. S Chult, Exploring network structure, dynamics, and function using NetworkX, Technical Report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.
40. O. Agarwal, H. Ge, S. Shakeri, R. Al-Rfou, Knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training, arXiv preprint