On Constructing Biomedical Text-to-Graph Systems with Large Language Models

Lorenzo Bertolini[1], Roel Hulsman[1], Sergio Consoli[1], Antonio Puertas-Gallardo[1] and Mario Ceresa[1]

[1]*European Commission, Joint Research Centre (JRC), Ispra, Italy*

Abstract

Knowledge graphs and ontologies represent symbolic and factual information that can offer structured and interpretable knowledge. Extracting and manipulating this type of information is a crucial step in complex processes such as human reasoning. While Large Language Models (LLMs) are known to be useful for extracting and enriching knowledge graphs and ontologies, previous work has largely focused on comparing architecture-specific models (e.g. encoder-decoder only) across benchmarks from similar domains. In this work, we provide a large-scale comparison of the performance of certain LLM features (e.g. model architecture and size) and task learning methods (fine-tuning vs. in-context learning (iCL)) on text-to-graph benchmarks in the biomedical domain. Our experiment suggests that, while a simple truncation-based heuristic can notably boost the performance of decoder-only models used with iCL, small fine-tuned encoder-decoder models produce the most stable and strong performance. Moreover, we found that a massive out-of-domain text-graph pre-training has a positive impact on fine-tuned models, while we observed only a marginal impact of pre-training and size for decoder-only iCL models.

Keywords

Information Extraction, Knowledge Graphs, Word Embeddings, Large Language Models, In-Context Learning

1. Introduction

Acquiring structured knowledge from text is a fundamental step in a complex process like reasoning and answering questions, whether such a process is carried out by a human or an artificial intelligence (AI) system [1]. In natural language processing (NLP), structured knowledge is often handled via ontologies or knowledge graphs [2, 3, 4]. Knowledge graphs are typically organised as collections of `[(head # relation # tail)]` triplets, such as `[(dog # isA # animal)]`, or `[(Rome # CapitalOf # Italy)]`. Knowledge graphs and ontologies play a pivotal role in representing knowledge across various domains, facilitating

Third International Workshop on Knowledge Graph Generation from Text, May 26-30, 2024, co-located with Extended Semantic Web Conference (ESWC), Hersonissos, Greece

lorenzo.bertolini@ec.europa.eu (L. Bertolini)

intelligent applications such as chatbots [5], recommendation systems [6] question answering systems [7, 8] and more [9, 1].

Knowledge graphs have seen a surge in their application in recent years [10, 11]. However, building them can be laborious and costly [8, 4]. This has led to the development of numerous methods aimed at auto-generation of these graphs from text sources in various fields [12, 9, 11, 4]. Until recently, extracting and manipulating knowledge graphs and other forms of graphs has been largely dealt with by small knowledge graph embedding models (KGEs) [13], which are lightweight but limited in capabilities, or different types of graph neural networks (GNNs) [14, 15], such as convolutional graph neural networks (CGNNs) [16], or gated attention graph neural networks (GAT-GNN) [17]. Recently, many of these architectures have been replaced by transformer-based large language models (LLMs) [18], which have shown great potential in modelling graph-based data.

Despite these advancements, current techniques still suffer from significant limitations concerning accuracy, completeness, privacy, bias, and scalability [19, 20, 4]. Therefore, generating a large-scale knowledge graph automatically from text corpora remains an open challenge [3, 9, 4]. As shown by a consistent body of evidence [21, 22, 23], LLMs can be adapted to both extract knowledge graphs from a reference text (text-to-graph task), as well as to convert knowledge graphs into natural language while maintaining the semantic meaning (graph-to-text task). We are interested in the former.

To adapt an LLM to a particular task, two popular task learning methods are fine-tuning and in-context learning (iCL) [24]. Given a training dataset pertaining to the new task at hand, fine-tuning an LLM amounts to an additional training phase to update a subset of learnable model parameters to adapt to the new task. In-context learning, on the other hand, consists of including a few task examples in the model prompt at inference time - a special case of few-shot learning. Typically, iCL provides weaker performance than fine-tuning and is computationally more expensive at inference time [25, 24], yet it is highly flexible as it does not require any parameter updates. Both options involve a vast amount of design choices, from the quality and quantity of available training data to the amount of in-context examples to include in iCL.

While most work on knowledge graph extraction has focused on pushing the state-of-the-art in terms of performance [26, 27] or summarising the field in terms of different applications [21, 22, 23] and formulations of scenarios and tasks [28, 29, 30], it remains unclear to the general AI practitioner what would be, given a specific dataset and computational resources, the best solution to approach a text-to-graph task, formulated as an end-to-end LLM-based solution.

This work is directed to the general AI practitioner in the biomedical domain aiming to develop an end-to-end LLM-based knowledge graph extraction system from textual sources. We investigate how to best approach such task by examining various combinations of model design choices, assuming a fixed and accessible computational resource of a single RTX 8000 GPU. The main variables under investigation are model architecture (encode-decoder, decoder-only), model family (T5, BART, Mistral-v0.1, Llama-2), model size (small (60M) to mid (13B learnable parameters)), task learning method (fine-tuning, iCL) and additional pre-training data (relation extraction data, conversation data, instruction data, (bio)medical data). In brief, the main insights of this paper encompass the following:

1. We provide tentative evidence that biomedical knowledge graphs can be hard to model. Mid-sized decoder-only models adopting iCL show weak performance, while performance of small fine-tuned encoder-decoder models is robust compared to the general domain.
2. For small fine-tuned encoder-decoder models we observe power-law scaling in model size, while for mid-sized decoder-only models adopting iCL we instead observe power-law scaling in the number of in-context examples. This is in line with known results [24].
3. Only additional pre-training data on relation extraction tasks boosts model performance, while neither observing conversation data, instruction data nor (bio)medical data during pre-training makes a notable difference.
4. We propose and experimentally prove the effectiveness of a simple truncation-based heuristic on model output to control for a specific type of hallucination of in-context learning, avoiding expensive prompt tuning and prompt design.

2. Material and Methods

Knowledge graph structure To ensure a stable and fair comparison across models with different pre-training, we pre-process the selected dataset to match the following linearised text-graph structure. Formally, a dataset consists of two sets of strings T and G, where each reference text $t_i \in T$ and knowledge graph $g_i \in G$ are assumed to be identical representations semantically, but differ syntactically. For example, given a reference text *"The pencil is on the table."*, we represent the corresponding knowledge graph as containing one linearised triplet "`[(pencil # IsOn # table)]`". In the coming paragraphs, we present a detailed example of the proposed linearisation, in the context of the prompt used for in-context-learning set-up.

Dataset We use BioEvent [30], a benchmark that aggregates 10 popular biomedical datasets, and adopt a simple strategy to clean up the data by removing any duplicate pairs and breaking ties in favour of the text-graph pair pertaining to the longest linearised knowledge graph, assuming that the longest knowledge graph is the most complete description of the entities and relations described, and finally obtain a train/validation/test set using an 80/10/10% split.

Metrics We evaluate performance with Rouge scores [31], namely Rouge-n ($n = 1, 2$) and Rouge-L. The former is based on n-grams, while the latter on the longest common sub-sequence (LCS) between two strings, as implemented by Hugging Face `evaluate` library.

Models We adopt models from two families of pre-trained encoder-decoder: *T5* and *BART*. We adopt three sizes in the T5 family, namely 60.5M parameters (`t5-small`), 223 M (`t5-base`) and 738 M (`t5-large`). As for BART, we use of the BART model (`bart-large`) introduced in [32], as well as a version from [33], tuned on REBEL dataset [34], a large relation extraction dataset designed for text-to-text modelling.

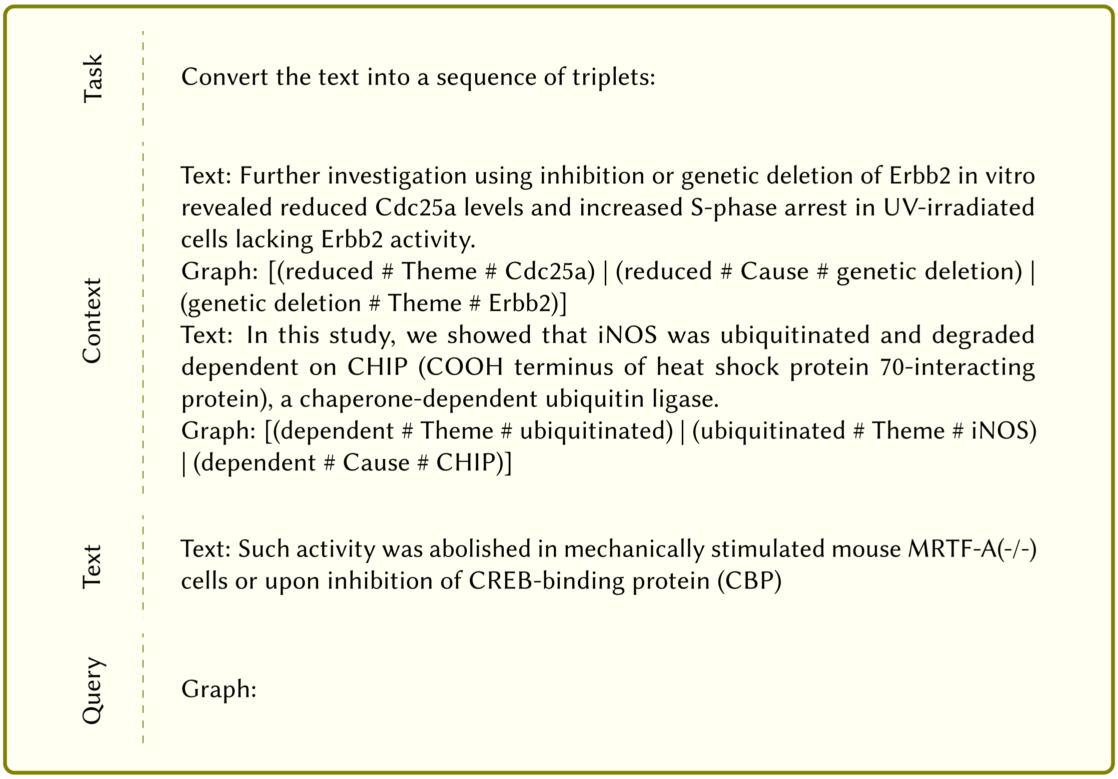
We then opt for two sets of pre-trained decoder-only LLMs: *Mistral-v0.1* and *Llama-2*. We include two sizes, one with 7B parameters (`Llama-2-chat-7b-hf`) and the largest model in our analysis with 13B parameters (`Llama-2-chat-13b-hf`), as well as a Llama-2 model fine-tuned on biomedical knowledge and question-answering (`meditron-7b`) [35], to investigate the beneficial effect of domain-specific training in the biomedical domain.

As for the Mistral family, we adopt three models. The original 7B model (`Mistral-v0.1`), a version fine-tuned on a variety of open-source conversation datasets (`Mistral-Instruct-v0.1`), and finally a version fine-tuned by OpenOrca (`Mistral-OpenOrca`) [36] on a reproduction attempt of the Orca dataset [37], leveraging the Flan Collection for effective instruction-tuning [38]. Importantly, all models adopted in this work are fully open-source and accessible through Hugging Face by adopting the transformer library [39].

Learning methods We adopt two distinct task learning methods, fine-tuning for the smaller encoder-decoder models and iCL [24] for the larger decoder-only models. All fine-tuning experiments are based on the `trainer` class implementation from Hugging Face. Given a text-graph pair (t_i, g_i) in the pre-defined training set, each model undergoes an additional fine-tuning phase where it is trained to generate the graph g_i as output, using the text t_i as input. All models are tuned end-to-end for up to ten epochs, selecting the best model based on the validation Rouge-1 score, as per standard practice in NLP and knowledge graph literature [40, 41]. For training, hyper-parameters are as given in Table 2 of Appendix A.

For the iCL setting, each pre-trained model is queried with a simple prompt, containing a set of N solved text-graph examples taken from the available training set. To limit the impact of selecting a set of poor examples, we sample N examples randomly from the training set for each test instance at inference time. Moreover, we omit time and computation-consuming prompt engineering and computationally expensive prompt tuning to resemble common practice of end-users. However, we highlight the importance of such practice to prevent model hallucinations, and, more generally, to prevent spurious features in prompt design along the lines of [42]. To provide a fair estimate of iCL performance, we introduce a simple *post-hoc hallucination-control heuristic* to determine the end of the desired structured output (i.e. the end of a knowledge graph). Simply put, we truncate model output at the appearance of the tokens "`)]`", signalling the end of a knowledge graph in our graph structure. An example of the finalised iCL prompt (with $N = 2$) is presented in Figure 1.

Experimental setup The main experiment is designed to unveil the approximate overall power of selected models and task learning methods, as well as to understand what impacts and shapes their performance. The models' algorithms have been implemented in Python version 3.8.5, and all the computations run on a single RTX 8000 GPU within a AMD EPYC 7282 16-Core 64-bit microprocessor at 1.50GHz with 512GB RAM. We do recognise that assuming larger computational power could significantly improve results, especially by including large decoder-only models or by fine-tuning the mid-sized Mistral-v0.1 and Llama-2 families, which is out of the computational reach of the current setup.



Task

Convert the text into a sequence of triplets:

Context

Text: Further investigation using inhibition or genetic deletion of Erbb2 in vitro revealed reduced Cdc25a levels and increased S-phase arrest in UV-irradiated cells lacking Erbb2 activity.
Graph: [(reduced # Theme # Cdc25a) | (reduced # Cause # genetic deletion) | (genetic deletion # Theme # Erbb2)]
Text: In this study, we showed that iNOS was ubiquitinated and degraded dependent on CHIP (COOH terminus of heat shock protein 70-interacting protein), a chaperone-dependent ubiquitin ligase.
Graph: [(dependent # Theme # ubiquitinated) | (ubiquitinated # Theme # iNOS) | (dependent # Cause # CHIP)]

Text

Text: Such activity was abolished in mechanically stimulated mouse MRTF-A(-/-) cells or upon inhibition of CREB-binding protein (CBP)

Query

Graph:

Figure 1: Example prompt - $N = 2$ in-context examples. The prompt ends with a query for a knowledge graph pertaining to the last reference text.

The main goal of the experiment is to understand how LLM characteristics and task-learning methods perform in our text-to-graph task, under fixed computational resources. Throughout, we aim to guide the general AI practitioner to understand which combination is most suited for such a task and to showcase how to navigate (part of) the vast and complex spectrum of model design choices. Given the fixed computational resources, we fine-tune the previously introduced set of smaller encoder-decoder models and compare performance to the set of larger decoder-only models in combination with iCL. This choice is framed in the context of a given computational resource such that fine-tuning is computationally infeasible for larger models. At the same time, the short context window of the T5 and BART families (1k tokens or below) proves iCL unsuitable. Following [43], we adopt $N = 8$ for the amount of in-context examples.

3. Results

The overall results of various combinations of model architecture, family, size, relevant pre-training data and task learning method are shown in Table 1. First, we can observe a clear benefit in fine-tuning smaller encoder-decoder models. We hypothesise this relates to issues regarding benchmark quality. Bio Event presents a high amount of unique entities and triplets, creating a complex distribution of patterns in reference texts that is difficult to infer correctly from just 8 in-context examples. Overall the best performance across metrics is reached with fine-tuned decoder-only models, i.e. the largest model in the T5 family.

Table 1
General experiment results. We report Rouge scores obtained by various combinations of model architecture, family, size (learnable parameters), relevant additional data seen during pre-training and our task learning method of fine-tuning or iCL.

					Bio Event		
Arch.	**Family**	**Size**	**Pre-training**	**Learning Method**	**R-1**	**R-2**	**R-L**
Encod.-decod.	T5	6.5M		Fine-tuning	.57	.42	.53
		223M		Fine-tuning	.62	.48	.59
		738M		Fine-tuning	.66	.54	.63
	BART	406M		Fine-tuning	.53	.38	.51
		406M	REBEL	Fine-tuning	.67	.56	.64
Decoder-only	Mistral-v.1	7B		iCL + Heuristic	.43	.25	.37
		7B		iCL	.16	.07	.13
		7B	Conversation	iCL + Heuristic	.43	.24	.36
		7B	Conversation	iCL	.18	.08	.14
		7B	OpenOrca	iCL + Heuristic	.44	.25	.36
		7B	OpenOrca	iCL	.24	.12	.19
	Llama-2	7B	Instruction	iCL + Heuristic	.44	.24	.37
		7B	Instruction	iCL	.17	.08	.14
		13B	Instruction	iCL + Heuristic	.44	.24	.37
		13B	Instruction	iCL	.16	.07	.13
		7B	Meditron	iCL + Heuristic	.40	.21	.34
		7B	Meditron	iCL	.17	.08	.14

Moreover, within both the T5 and Llama-2 family, we find a clear positive correlation between model size and performance. This is in line with the well-documented phenomenon of power-law scaling of LLM performance in the number of model parameters [44]. Focusing on the BART family, we see that adopting an additional relation extraction dataset during pre-training (REBEL) yields universally superior results. This is in sharp contrast to other pre-training additions, since neither conversation data nor instruction, OpenOrca or Meditron datasets seem to affect performance on either benchmark. We hypothesise none are particularly relevant to our text-to-graph task, although this is notably most surprising for the biomedical knowledge in the Meditron pre-training data.

Table 1 also shows that our hallucination-control heuristic for iCL models yields a large performance boost, independently of architecture, family, size, or pre-training data. To briefly reiterate, this was put in place to avoid computationally and experimentally demanding prompt engineering or tuning, and implemented by truncating model output after tokens signalling a graph's end (i.e., “)] ”). The jump in performance can reach more than 20 points, and is consistent across all Rouge scores. Concerning specific metrics, we found Rouge-1 (R-1) scores to be consistently higher, especially in decoder-only models, indicating a stronger entity and relation recognition. We also found R-L scores to be systematically above Rouge-2 (R-2) score, and closer to R-1. This suggests that the identified entities and relations are often in the right order, but certain entities or relations are missing such that correct 2-grams are lacking.

4. Discussion

This work is directed at biomedical researchers and practitioners aiming to develop an end-to-end LLM-based automatic graph extraction system from textual sources. Assuming a realistic computational baseline, our large-scale comparison contributed to the development of a more effective and efficient pipeline for biomedical knowledge extraction and representation tasks by highlighting the impact of a plethora of design choices and provided several empirical insights.

Indeed, off-the-shelf LLMs together with a task learning method can achieve strong entity and relation recognition, and reach moderate yet promising overall results on knowledge graph completion. The optimal performance of LLMs is likely higher than displayed here, e.g. due to prompt engineering/tuning, hyper-parameter tuning, more computational power and more model parameters. Our results indicate that, without fine-tuning, LLMs might not be directly suitable for biomedical text-to-graph tasks. Fine-tuning has proven more robust than iCL, since mid-sized decoder-only models adopting iCL show weak performance, while small fine-tuned encoder-decoder models achieve robust moderate results. We hypothesise that expert knowledge contained in reference texts in the biomedical domain poses a more difficult knowledge extraction problem, such that iCL with a small amount of in-context examples is not sufficient to correctly extrapolate said task. That is, knowledge graphs in the biomedical domain might require knowledge obtained across a large set of examples. However, we provide strong and consistent evidence for our simple truncation-based heuristic to be highly effective in boosting model performance without time-expensive prompt engineering and computationally expensive prompt tuning, which is not necessarily generalisable across subsets of the same task [45]. Crucially, this suggests that when the output of a model follows a constrained structure, simple rule-based heuristics can be an efficient method to limit undesired output.

5. Conclusions

This work examined the ability of LLMs to generate biomedical knowledge graphs from reference texts, comparing end-to-end fine-tuned encoder-decoder models, against decoder-only models used with in-context learning (iCL). Our results showed how small fine-tuned encoder-decoder models consistently outperform mid-sized decoder-only models adopting iCL. We found evidence that our simple heuristic to control for model hallucination has a consistently positive impact on the performance of decoder-only models, but no connection between performance and including additional datasets during pre-training that are not directly linked to the text-to-graph task, such as conversation-tuning, instruction-tuning and biomedical expert knowledge. On the contrary, we found that including a relation-extracting dataset like REBEL showed a notable boost in the performance of encoder-decoder models, for which we also observed a power-law connection between model size and performance.

References

[1] S. Tiwari, F. Ortíz-Rodriguez, S. B. Abbés, P. U. Usip, R. Hantach, Semantic AI in Knowledge Graphs, Taylor & Francis, Boca Raton, US, 2023. doi:10.1201/9781003313267.

[2] H. Paulheim, Knowledge graph refinement: A survey of approaches and evaluation methods, Semantic Web 8 (2017).

[3] A. Hogan, E. Blomqvist, M. Cochez, C. D'Amato, G. D. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, A.-C. N. Ngomo, A. Polleres, S. M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, A. Zimmermann, Knowledge graphs, ACM Computing Surveys 54 (2021). doi:10.1145/3447772.

[4] C. Peng, F. Xia, M. Naseriparsa, F. Osborne, Knowledge Graphs: Opportunities and Challenges, Artificial Intelligence Review 56 (2023). doi:10.1007/s10462-023-10465-9.

[5] A. Ait-Mlouk, L. Jiang, KBot: A Knowledge Graph Based ChatBot for Natural Language Understanding over Linked Data, IEEE Access 8 (2020). doi:10.1109/ACCESS.2020.3016142.

[6] Y. Xian, Z. Fu, S. Muthukrishnan, G. De Melo, Y. Zhang, Reinforcement Knowledge Graph Reasoning for Explainable Recommendation, Association for Computing Machinery, New York, NY, USA, 2019. doi:10.1145/3331184.3331203.

[7] X. Huang, J. Zhang, D. Li, P. Li, Knowledge Graph Embedding Based Question Answering, Association for Computing Machinery, New York, NY, USA, 2019. doi:10.1145/3289600.3290956.

[8] M. Kejriwal, J. Sequeda, V. Lopez, Knowledge Graphs: Construction, Management and Querying, Semantic Web 10 (2019). doi:10.3233/SW-190370.

[9] M. Kejriwal, Knowledge Graphs: A Practical Review of the Research Landscape, Information 13 (2022). doi:10.3390/info13040161.

[10] X. Chen, S. Jia, Y. Xiang, A Review: Knowledge Reasoning Over Knowledge Graph, Expert Systems with Applications 141 (2020). doi:10.1016/j.eswa.2019.112948.

[11] S. Ji, S. Pan, E. Cambria, P. Marttinen, P. S. Yu, A Survey on Knowledge Graphs: Representation, Acquisition, and Applications, IEEE Transactions on Neural Networks and Learning Systems 33 (2022). doi:10.1109/TNNLS.2021.3070843.

[12] Q. Liu, Y. Li, H. Duan, Y. Liu, Z. Qin, Knowledge graph construction techniques, Journal of Computer Research and Development 53 (2016). doi:10.7544/issn1000-1239.2016.20148228.

[13] Q. Wang, Z. Mao, B. Wang, L. Guo, Knowledge Graph Embedding: A Survey of Approaches and Applications, IEEE Transactions on Knowledge and Data Engineering 29 (2017). doi:10.1109/TKDE.2017.2754499.

[14] Z. Ye, Y. J. Kumar, G. O. Sing, F. Song, J. Wang, A comprehensive survey of graph neural networks for knowledge graphs, IEEE Access 10 (2022). doi:10.1109/ACCESS.2022.3191784.

[15] L. Wu, Y. Chen, K. Shen, X. Guo, H. Gao, S. Li, J. Pei, B. Long, Graph Neural Networks for Natural Language Processing: A Survey, Foundations and Trends in Machine Learning 16 (2023). doi:10.1561/2200000096.

[16] S. Zhang, H. Tong, J. Xu, R. Maciejewski, Graph Convolutional Networks: Algorithms, Applications and Open Challenges, Springer International Publishing, Cham, 2018. doi:10.

1007/978-3-030-04648-4_7.

[17] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, Graph Attention Networks, in: International Conference on Learning Representations, 2018. URL: https://openreview.net/forum?id=rJXMpikCZ.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, Advances in Neural Information Processing Systems 30 (2017).

[19] F. Radulovic, N. Mihindukulasooriya, R. García-Castro, A. Gómez-Pérez, A comprehensive quality model for linked data, Semantic Web 9 (2018). doi:10.3233/SW-170267.

[20] M. R. A. Rashid, G. Rizzo, M. Torchiano, N. Mihindukulasooriya, O. Corcho, R. García-Castro, Completeness and consistency analysis for evolving knowledge bases, Journal of Web Semantics 54 (2019). doi:10.1016/j.websem.2018.11.004.

[21] B. Jin, G. Liu, C. Han, M. Jiang, H. Ji, J. Han, Large language models on graphs: A comprehensive survey, arXiv preprint arXiv:2312.02783 (2023).

[22] J. Liu, C. Yang, Z. Lu, J. Chen, Y. Li, M. Zhang, T. Bai, Y. Fang, L. Sun, P. S. Yu, et al., Towards graph foundation models: A survey and beyond, arXiv preprint arXiv:2310.11829 (2023).

[23] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang, X. Wu, Unifying large language models and knowledge graphs: A roadmap, IEEE Transactions on Knowledge and Data Engineering (2024). doi:10.1109/TKDE.2024.3352100.

[24] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, Advances in Neural Information Processing Systems 33 (2020). URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[25] H. Liu, D. Tam, M. Muqeeth, J. Mohta, T. Huang, M. Bansal, C. A. Raffel, Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning, Advances in Neural Information Processing Systems 35 (2022).

[26] Q. Guo, Z. Jin, X. Qiu, W. Zhang, D. Wipf, Z. Zhang, CycleGT: Unsupervised graph-to-text and text-to-graph generation via cycle training, in: T. Castro Ferreira, C. Gardent, N. Ilinykh, C. van der Lee, S. Mille, D. Moussallem, A. Shimorina (Eds.), Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+), Association for Computational Linguistics, Dublin, Ireland (Virtual), 2020. URL: https://aclanthology.org/2020.webnlg-1.8.

[27] Z. Jin, Q. Guo, X. Qiu, Z. Zhang, GenWiki: A dataset of 1.3 million content-sharing text and graphs for unsupervised graph-to-text generation, in: D. Scott, N. Bel, C. Zong (Eds.), Proceedings of the 28th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Barcelona, Spain (Online), 2020. doi:10.18653/v1/2020.coling-main.217.

[28] L. Wang, Y. Li, O. Aslan, O. Vinyals, WikiGraphs: A Wikipedia text - knowledge graph paired dataset, in: A. Panchenko, F. D. Malliaros, V. Logacheva, A. Jana, D. Ustalov, P. Jansen (Eds.), Proceedings of the Fifteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-15), Association for Computational Linguistics,

Mexico City, Mexico, 2021. doi:10.18653/v1/2021.textgraphs-1.7.

[29] A. Colas, A. Sadeghian, Y. Wang, D. Z. Wang, Eventnarrative: A large-scale event-centric dataset for knowledge graph-to-text generation, in: Thirty-fifth Conference on Neural Information Processing (NeurIPS 2021) Track on Datasets and Benchmarks, 2021.

[30] G. Frisoni, G. Moro, L. Balzani, Text-to-text extraction and verbalization of biomedical event graphs, in: Proceedings of the 29th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Gyeongju, Republic of Korea, 2022. URL: https://aclanthology.org/2022.coling-1.238.

[31] C.-Y. Lin, ROUGE: A package for automatic evaluation of summaries, in: Text Summarization Branches Out, Association for Computational Linguistics, Barcelona, Spain, 2004. URL: https://www.aclweb.org/anthology/W04-1013.

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

[33] G. Rossiello, M. F. M. Chowdhury, N. Mihindukulasooriya, O. Cornec, A. M. Gliozzo, Knowgl: Knowledge generation and linking from text, in: The Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI Press, 2023, pp. 16476–16478. doi:10.1609/aaai.v37i13.27084.

[34] P.-L. Huguet Cabot, R. Navigli, REBEL: Relation extraction by end-to-end language generation, in: Findings of the Association for Computational Linguistics: EMNLP 2021, Association for Computational Linguistics, Punta Cana, Dominican Republic, 2021. URL: https://aclanthology.org/2021.findings-emnlp.204.

[35] Z. Chen, A. H. Cano, A. Romanou, A. Bonnet, K. Matoba, F. Salvi, M. Pagliardini, S. Fan, A. Köpf, A. Mohtashami, et al., Meditron-70b: Scaling medical pretraining for large language models, arXiv preprint arXiv:2311.16079 (2023).

[36] W. Lian, B. Goodson, G. Wang, E. Pentland, A. Cook, C. Vong, "Teknium", MistralOrca: Mistral-7B Model Instruct-tuned on Filtered OpenOrcaV1 GPT-4 Dataset, HuggingFace repository (2023).

[37] S. Mukherjee, A. Mitra, G. Jawahar, S. Agarwal, H. Palangi, A. Awadallah, Orca: Progressive learning from complex explanation traces of gpt-4, arXiv preprint arXiv:2306.02707 (2023).

[38] S. Longpre, L. Hou, T. Vu, A. Webson, H. W. Chung, Y. Tay, D. Zhou, Q. V. Le, B. Zoph, J. Wei, et al., The flan collection: Designing data and methods for effective instruction tuning, arXiv preprint arXiv:2301.13688 (2023).

[39] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, A. Rush, Transformers: State-of-the-art natural language processing, in: Q. Liu, D. Schlangen (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, 2020. doi:10.18653/v1/2020.emnlp-demos.6.

[40] I. Balazevic, C. Allen, T. Hospedales, Multi-relational poincaré graph embeddings, Advances in Neural Information Processing Systems 32 (2019). URL: https://proceedings.

neurips.cc/paper_files/paper/2019/file/f8b932c70d0b2e6bf071729a4fa68dfc-Paper.pdf.

[41] I. Chami, A. Wolf, D.-C. Juan, F. Sala, S. Ravi, C. Ré, Low-dimensional hyperbolic knowledge graph embeddings, in: D. Jurafsky, J. Chai, N. Schluter, J. Tetreault (Eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2020. doi:10.18653/v1/2020.acl-main.617.

[42] M. Sclar, Y. Choi, Y. Tsvetkov, A. Suhr, Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting, arXiv preprint arXiv:2310.11324 (2023).

[43] J. Wei, X. Wang, D. Schuurmans, M. Bosma, b. ichter, F. Xia, E. Chi, Q. V. Le, D. Zhou, Chain-of-thought prompting elicits reasoning in large language models, Advances in Neural Information Processing Systems 35 (2022). URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf.

[44] J. Hestness, S. Narang, N. Ardalani, G. Diamos, H. Jun, H. Kianinejad, M. M. A. Patwary, Y. Yang, Y. Zhou, Deep learning scaling is predictable, empirically, arXiv preprint arXiv:1712.00409 (2017).

[45] L. Bertolini, J. Weeds, D. Weir, Testing large language models on compositionality and inference with phrase-level adjective-noun entailment, in: N. Calzolari, C.-R. Huang, H. Kim, J. Pustejovsky, L. Wanner, K.-S. Choi, P.-M. Ryu, H.-H. Chen, L. Donatelli, H. Ji, S. Kurohashi, P. Paggio, N. Xue, S. Kim, Y. Hahm, Z. He, T. K. Lee, E. Santus, F. Bond, S.-H. Na (Eds.), Proceedings of the 29th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Gyeongju, Republic of Korea, 2022. URL: https://aclanthology.org/2022.coling-1.359.

A. Fine-tuning hyper-parameters

Hyperparameters set, with their respective values adopted for our experiments with the encoder-decoder models.

Table 2
Hyper-parameters used in fine-tuning. Where unspecified, default values in Hugging Face's `trainer` class apply.

Hyper-parameter	**Value**
Seed	42
Evaluation Strategy	epoch
Epochs	10
Warm-up steps	10
Validation metric	eval_rouge1
Calculate generative metrics (i.e. Rouge)	True
Optimizer	AdamW
Learning rate	5e-05
Weight decay	0.01
ADAM β_1	0.9
ADAM β_2	0.999
ADAM ϵ	1e-0.8
Label smoothing factor	0.1
Train batch size	24
Validation batch size	24
Group samples of similar length	True
16-bit precision training	True