On General and Biomedical Text-to-Graph Large Language Models

Lorenzo Bertolini[†] [*], Roel Hulsman[†], Sergio Consoli, Antonio Puertas Gallardo and Mario Ceresa
European Commission, Joint Research Centre (JRC), Ispra, Italy
E-mail: lorenzo.bertolini@ec.europa.eu

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. 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 two domains, namely the general and biomedical ones. Experiments suggest that, in the general domain, small fine-tuned encoder-decoder models and mid-sized decoder-only models used with iCL reach overall comparable performance with high entity and relation recognition and moderate yet encouraging graph completion. Our results further tentatively suggest that, independent of other factors, biomedical knowledge graphs are notably harder to learn and better modelled by small fine-tuned encoder-decoder architectures. Pertaining to iCL, we analyse hallucinating behaviour related to sub-optimal prompt design, suggesting an efficient alternative to prompt engineering and prompt tuning for tasks with structured model output.

Keywords: knowledge graphs, automatic graph generation, large language models, in-context learning, biomedical NLP

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 [2–4]. 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 intelligent applications such as chatbots [5], recommendation systems [6] question-answering systems [7, 8] and more [1, 9].

Knowledge graphs have seen a surge in their application in recent years [10, 11]. However, building them can be laborious and costly [4, 8]. This has led to the development of numerous methods aimed at auto-generation of these graphs from text sources in various fields [4, 9, 11, 12]. Until recently, extracting and manipulating knowledge graphs and other forms of graphs has been largely dealt with by small knowledge graph embedding models (KGEs)

[*]Corresponding author. E-mail: lorenzo.bertolini@ec.europa.eu.
[†]Equal contribution.
Code available: https://github.com/jrcf7/txt2graphLLMs.

[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 [4, 19, 20]. Therefore, generating a large-scale knowledge graph automatically from text corpora remains an open challenge [3, 4, 9]. As shown by a consistent body of evidence [21–23], 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 and adopt two text-to-graph benchmark datasets, to be referred to as Web NLG [24] and Bio Event [25]. Web NLG is a popular benchmark containing text-graph pairs of multiple types or relations, maintaining a rather general domain, while Bio Event pertains to biomedical data by aggregating 10 popular biomedical datasets.

To adapt an LLM to a particular task, two popular task learning methods are fine-tuning and in-context learning (iCL) [26]. 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, amounts to 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 [26, 27], 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 or summarising the field in terms of different applications and formulations of scenarios and tasks, 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.

1.1. Research objective and contribution

We direct this work to the general AI practitioner in the general or 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 NVIDIA Quadro RTX 8000 GPU[1]. 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. In the general domain, we show that small fine-tuned encoder-decoder models and mid-sized decoder-only models adopting iCL achieve comparable results with high entity and relation recognition and moderate yet encouraging graph completion.
2. We provide tentative evidence that biomedical knowledge graphs are substantially harder to model from textual sources than the general domain. 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. However, we discover issues with the biomedical benchmark adopted from [25] and a thorough revision is required to adopt it as gold standard for text-to-graph tasks.

[1]For reproducibility on less powerful hardware, we suggest optimizing performance through techniques such as quantization, Low-Rank Adaptation of Large Language Models (LoRA), knowledge distillation, or pruning, or utilizing cloud-based solutions, such as GPU-accelerated cloud services or model serving platforms, to make our approach more accessible to a broader range of practitioners. These strategies could help alleviating the computational resource constraints and make our approach more accessible to a broader range of practitioners.

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.
5. Off-the-shelve LLMs in the zero-shot setting (i.e. no in-context examples, only a task instruction and reference text) show weak performance. Careful design choices and a task learning method are required to be suitable for such task, especially in safety-critical and domain-specific contexts such as the biomedical domain. We highlight several areas of importance.

1.2. Paper structure

This paper is structured as follows. Section 2 introduces related work, with a focus on architecture, tasks, and proposed benchmarks. Subsequently, Section 3 presents methodology, including datasets (3.2), metrics (3.3), model architectures (3.4), task learning methods (3.5), and experiments' set-up (3.6). Then Section 4 shows experimental results, followed by Section 5 and 6 to respectively discuss and conclude.

2. Related Work

Recent surveys suggest LLMs are of primary interest and hold potential for multiple types of graph-based tasks [21–23, 28]. The task of automatically generating a knowledge graph from a reference text (text-to-graph) is closely related to the more general NLP task of relation extraction, traditionally composed of the two separated steps of named entity recognition and relation classification [29]. Named entity recognition involves identifying and classifying entities in the text, which can be seen as nodes in the resulting knowledge graph [30]. Sub-tasks include co-reference resolution and entity disambiguation. Relation classification, in turn, aims at identifying the relation between two given entities, as (often implicitly) expressed in the reference text containing the identified entities. In an LLM-based knowledge graph extraction system, both steps are potentially entangled in a single end-to-end solution.

Previous works have explored different approaches to text-to-graph tasks and the utilisation of LLMs [31–33]. However, the main focus has been on pushing the state of the art on specific sub-tasks and benchmarks, such as research data [34–36], question-answering [37, 38], common-sense [39, 40], biomedical [41, 42], and other [43]. Moreover, unlike this work, the literature does not offer a systematic experimental comparison of the efficiency of contributing factors in an end-to-end text-to-graph task, as suggested by the summary proposed in Table 1. For instance, Bosselut et al. [44] proposed COMET, a model that generates commonsense knowledge graphs from textual inputs. At the same time, the authors introduced the ATOMIC dataset, which is designed for commonsense inference, and utilised a transformer-based model to extract and generate the graph-based knowledge. The presented transformer model was only compared against existing LSTM-based solutions, which were already known to be subsumed by transformers.

Further work on the state of the art includes Guo et al. [45] introducing CycleGT, a two-loss model to learn from text-to-graph and graph-to-text tasks, based on an encoder-decoder pre-trained model (T5), by bootstrapping from fully non-parallel graph and text data, and iteratively back translating between the two forms. The authors propose a comparison of this solution and architecture on multiple general-domain datasets, using alignment and an unsupervised setting [52]. In addition, Dash et al. [46] propose a new text-to-graph model, called CUVA (Canonicalizing Using Variational Autoencoders), that addresses the redundancy and ambiguity of noun and relation phrases in open knowledge graphs. Unlike current methods that use a two-step process, CUVA simultaneously learns both embeddings and cluster assignments, resulting in better performance. Additional advancements in the field of knowledge graph extraction include Zhang et al. [47], who demonstrate the effectiveness of pre-trained models for generating knowledge graphs from text when fine-tuned with graph-aware objectives. They propose a graph-augmented text representation model that significantly improves the performance of this task.

Table 1

Related work summary. Schematic representation of the most relevant related work on text-to-graph, focusing on 5 dimensions: Model architecture (Encoder-Decoder vs. Decoder-Only), main benchmark (name), Learning method (tuning vs. in-context learning (iCL)), and new (yes vs. no, for both model architecture and benchmark).

Work	Model Architecture	New	Main Benchmarks	New	Learning Method
[44]	Decoder-only	Yes	Atomic	No	Fine-tuning
[45]	Encoder-decoder	Yes	WebNLG	No	Fine-tuning
[46]	Variational autoencoder	Yes	CANONICNELL	No	Fine-tuning
[47]	Encoder-only	Yes	IMDB, Yelp	No	Fine-tuning
[48]	Encoder-decoder	No	Wikigraphs	Yes	Fine-tuning
[49]	Encoder-decoder	No	EventNarrative	Yes	Fine-tuning
[50]	Decoder-only	No	Text2KGBench	Yes	iCL
[51]	Decoder-only	No	Text2KGBench	No	zero-shot
Ours	Encoder-decoder, Decoder-only	No	WebNLG. Bioevent	No	Fine-tuning, iCL

Other relevant work has focused on introducing new datasets and benchmarks. Wang et al. [48] introduce Wiki Graphs, a dataset to benchmark text-to-graph and graph-to-text tasks. The study focuses on a single solution, a combination of GNN graph-transformers, compared against transformer models. The study then focuses more on the structure of the graph than its content and, as such, the dataset is generally stripped of the name entity in the task outputs. In Colas et al. [49], the authors introduce Event Narrative, an event-based text-graph and graph-to-text dataset. Similar to previous endeavours, the authors compare a graph-based transformer with two encoder-decoder pre-trained LLMs, namely T5 and BART, and find mixed performance. Frisoni et al. [25] test a self-implemented version of these models on a new benchmark of biomedical graph-to-text and text-to-graph datasets.

In another study, Mihindukulasooriya et al. [50] introduced a text-to-graph dataset, *guided* by another ontology. The authors tested their methods using a combination of prompt generation, pre-trained decoder-only LLMs, and post-processing. Our goal is similar, yet notably different, in that we aim to test how to use an *end-to-end* solution, starting solely from text, and requiring no post-processing.

The previous paragraphs suggest a disparity in the proposed approaches to the task under investigation, in terms of a lack of decoder-only solutions. Of note, Khorashadizadeh et al. [51] propose a qualitative analysis of the abilities of multiple high-level and mostly decoder-only models, such as ChatGPT and Bard (now Gemini), in terms of knowledge graph completion and question answering. Focusing on biomedical queries, the authors conclude that ChatGPT might present a valuable asset in automatically extracting knowledge graphs, albeit at a significant computational cost. Hu et al. [53], examined the impact of incorporating graph structural information into the encoding process of a decoder-only model. Their proposed model, GPT-GNN, combines the generative pre-trained transformer with graph neural networks to enhance the learning of graph representations.

To the best of our knowledge, this work proposes a first *quantitative* investigation of the abilities of both encoder-decoder and decoder-only models, without the aid of any prompt-construction resource. Compared to [51], our work offers a deeper and more broad understanding of the iCL capacities of LLMs, under different amounts of in-context examples and dataset domains, while also including a selection of *qualitative* examples (see Appendix A), showcasing the strengths and failures of these tools. On the other hand, our work offers an analysis of how these models behave under simple prompts, with respect to task recognition and hallucination, and offers a simple and cost-efficient solution to control them.

Our work differs from previous studies in knowledge graph extraction in that most proposed either a new model, dataset, or comparison between relatively similar transformer models, while this study experimentally compares the effect of model architectures, family, size and pre-training data across two different domains, namely general and biomedical. As such the goal is not to push state-of-the-art performance, but to study the impact of each factor and describe best practices and important considerations, given a specific amount of computational resources.

3. Material and Methods

Here we discuss materials and methods, focusing on knowledge graph structure (Section 3.1), benchmark datasets (Section 3.2), evaluation metrics (Section 3.3), LLMs of various architecture, family, size and pre-training characteristics (Section 3.4), task learning methods (Section 3.5) and experimental setup (Section 3.6).

3.1. Knowledge graph structure

To ensure a stable and fair comparison across domains, we pre-process our benchmark datasets 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 general, a knowledge graph g_i containing n triplets `(head # relation # tail)` follows the structure

$$g_i = \text{"[(head}_1\ \#\ \text{relation}_1\ \#\ \text{tail}_1)\ |\ \ldots\ |\ (\text{head}_n\ \#\ \text{relation}_n\ \#\ \text{tail}_n)]\text{"}$$

3.2. Benchmark datasets

We adopt two parallel text-to-graph datasets, to be referred to as Web NLG [24] and Bio Event [25]. Table 2 contains basic descriptive statistics for both datasets.

Table 2
Descriptive statistics of the Web NLG and Bio Event dataset.

		Web NLG	**Bio Event**
Dataset size	Train set	13 211	18 417
	Validation set	1 667	2 302
	Test set	1 779	2 302
	seen categories	966	
	unseen categories	813	
Descriptive statistics	# unique entities	3 605	19 706
	# unique relations	411	38
	# unique triplets	4 351	50 939
	avg. #tokens per text (s.d.)	23 (12)	30 (15)
	avg. #tokens per graph (s.d.)	34 (18)	29 (19)
	avg. #triplets per graph (s.d.)	2.93 (1.53)	3.23 (2.10)
Atom divergence	train-validation	0.06	0.16
	train-test	0.59	0.16
	train-test (seen categories)	0.54	
	train-test (unseen categories)	0.83	
Compound divergence	train-validation	0.05	0.66
	train-test	0.69	0.66
	train-test (seen categories)	0.50	
	train-test (unseen categories)	0.99	

The Web NLG dataset (we adopt version 3.0) is a widely used text-to-graph dataset that contains text-graph pairs of multiple types or relations, maintaining a rather general domain. For each text-graph pair, the corresponding DBpedia category is available, pertaining to the topic of the Wikipedia article it is extracted from. Using this information, we divide the test set into categories that are either seen or unseen in the training and validation data, providing opportunity to test in- and out-of-distribution generalisation capabilities of LLMs (see Appendix C). Out of a total of 18 categories, the categories film, scientist and musical work are unseen in training and validation.

Contrary to the general domain, Bio Event pertains to biomedical data and aggregates 10 popular biomedical datasets, thus (potentially) representing an important domain-specific benchmark for text-to-graph tasks in healthcare. However, the overall quality of the Bio Event dataset requires further comments. Originally presented in [25] as separate datasets for text-to-graph and graph-to-text tasks, we adopt the graph-to-text dataset for our text-to-graph task as it surprisingly includes a larger amount of unique text-graph pairs. Another surprising finding is how in this Bio Event dataset for the graph-to-text task, there are up to 78 unique sets of triplets corresponding to a single reference text. This contradicts the underlying assumption that knowledge graphs and reference texts only differ syntactically and not semantically. While the same knowledge graphs could correspond to multiple natural language reference texts with the same semantic meaning, a reference text naturally only has one corresponding knowledge graph that contains the full set of entities and relations described. Finally, there exists a large degree of dataset contamination, where the same reference text appears in both train and test set, yet connected to a different knowledge graph.

For the purpose of this work, we opt for a quick heuristic to clean up the Bio Event dataset in order to use it as a biomedical alternative to the general domain of Web NLG. First, we refine the Bio Event graph-to-text dataset by removing any duplicate reference texts 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. This filters out 66% of the datapoints in the original Bio Event dataset, explained by the earlier observation that a single reference text in that dataset often corresponds to multiple unique sets of triplets, whereas we limit ourselves to strictly unique text-graph pairs. We further process knowledge graphs to match the setup of Web NLG by removing meta-data associated with nodes and edges and finally obtain a train/validation/test set using a 80/10/10% split.

The issue remains that we observe in Table 2 a large number of unique entities and triplets compared to the amount of available examples. Furthermore, manual inspection (see Appendix A for cherry-picked examples) shows that reference texts and knowledge graphs do not always contain the same set of entities and relations and thus differ semantically. For example, consider the following tuple taken from the Bio Event test set,

$$(t_i, g_i) = \big(\text{“The semaphorin 7A receptor Plexin C1 is lost during melanoma metastasis.”},\ \text{“}\texttt{[(metastasis \# Theme \# melanoma)]}\text{”}\big).$$

Producing the ‘correct’ knowledge graph in this instance requires (i) ignoring most of the text, and (ii) producing a relation that is absent in any grammatical form. The opposite is true for general domain graphs in Web NLG, which contain relations such as `Owner` or `CompletionDate`, which are notably more transparent and frequently explicitly named in the reference text. This phenomenon is reflected in the descriptive statistics in Table 2. We observe that the average number of tokens per reference text in Bio Event is larger than in Web NLG (30 vs 23), while the average number of tokens per graph is lower (29 vs 34). This implies that relative to the Web NLG benchmark, Bio Event has smaller graphs corresponding to its reference texts, suggesting a problem of type (i). In addition, the number of unique relations in the triplets in the Bio Event benchmark is low compared to Web NLG (38 vs 411), while intuition suggests the reference texts in the biomedical domain contain more sophisticated domain language and thus a wider spectrum of relations, suggesting a problem of type (ii). We conclude that Bio Event does not classify as gold standard for a text-to-graph task and we encourage future work to come up with a thorough revision. Results on our refined Bio Event dataset should thus be regarded as tentative on the biomedical domain.

Finally, we follow [54] in assessing the compositional generalisation aspect of the train, validation and test set. In brief, text, or knowledge graphs, in this case, can be categorised into *atoms*, and and *compounds*. Atoms refer to atomic instances of the text, such as single word and grammatical or syntactic rules, whereas compounds are the ways single atoms are combined, e.g. fully formed sentences. In our work, atoms indicate heads, relations and tails, while compounds refer to combinations of atoms in the form of triplets. The divergence in atom and compound empirical distributions between datasets provides an estimate of how well our experiments adhere to the principles of compositional generalisation, as framed in [54]. That is, the divergence between distributions assesses whether our train-test experiments present a challenge from a compositional generalisation perspective, evidenced by high

compound divergence between train and test sets, while exclusively measuring the recombination of known atoms into compounds through low atom divergence.

On Web NLG, we observe that the train and validation set are similar in terms of both atoms and compounds, whereas the train and test set differ substantially in both atom and compound divergence. Unsurprisingly, this is highest for unseen categories. How the distribution of Web NLG examples over train and test set originates is unclear, but a redistribution of examples over train, validation and test set in future work could be beneficial to more robustly test compositional generalisation. On Bio Event, we see low atom divergence and high compound divergence between the train, validation and test set, in line with the principles of compositionality. We leave the general practitioner with the recommendation to pay attention to the quality of benchmark datasets, e.g. in terms of descriptive statistics and atom/compound divergence, as per the common intuition that model performance is highly sensitive to the quality of the underlying data.

3.3. Evaluation metrics

We evaluate a model's performance with Recall-Oriented Understudy for Gisting Evaluation (Rouge) scores [55]. Originally introduced for summarisation evaluation, the set of metrics is transferable to our text-to-graph setup by identifying a graph as a single-sentence summary. We specifically focus on Rouge-n ($n = 1, 2$) and Rouge-L, where the former is based on n-grams and the latter on the longest common sub-sequence (LCS) between two strings.

All scores are reported as the harmonic mean between recall and precision, making use of the implementation by Hugging Face[2],[3]. The exact formulas to calculate Rouge-n and Rouge-L over a set of candidate and reference graphs $(g_{\text{cand}}, g_{\text{ref}}) \in \mathcal{G}$ are given by

$$\text{Rouge-}n = \frac{1}{|\mathcal{G}|} \sum_{(g_{\text{cand}}, g_{\text{ref}}) \in \mathcal{G}} \frac{2\left|n\text{-gram}(g_{\text{cand}}) \cap n\text{-gram}(g_{\text{ref}})\right|}{\left|n\text{-gram}(g_{\text{cand}})\right| + \left|n\text{-gram}(g_{\text{ref}})\right|}, \tag{1}$$

$$\text{Rouge-L} = \frac{1}{|\mathcal{G}|} \sum_{(g_{\text{cand}}, g_{\text{ref}}) \in \mathcal{G}} \frac{2\left|\text{LCS}(g_{\text{cand}}, g_{\text{ref}})\right|}{\left|1\text{-gram}(g_{\text{cand}})\right| + \left|1\text{-gram}(g_{\text{ref}})\right|}, \tag{2}$$

where n-gram$(\cdot)$ returns the set of n-gram tokens in a graph, LCS$(\cdot, \cdot)$ returns the set of tokens of the longest common sub-sequence between two graphs and $|\cdot|$ returns the cardinality of a set.

The classical knowledge graph extraction pipeline consists of multiple stages, typically including co-reference resolution, named entity recognition, entity disambiguation and relationship classification. Using LLMs for knowledge graph extraction entangles all stages into one, therefore complicating the process of ascribing model errors to one or more stages. Comparing the Rouge metrics employed here, however, allows some insights. For example, Rouge-1 is a direct measure of entity and relation recognition, although it does not distinguish between either. Furthermore, Rouge-2 and Rouge-L additionally take sequence order into account, such that the difference with Rouge-1 is a measure of entities and relation being in the right order. In both cases, however, entity ambiguity and co-reference problems might be confounding factors. Finally, we note that the longest common sub-sequence differs from the

[2]We note three important design choices in the wrapper of Hugging Face and the underlying implementation of Rouge metrics by Google Research. First, Google Research removes all non-alpha-numeric characters and tokens are identified through space separation, thus removing graph tokens. Second, the Rouge-n F1 score reported by Hugging Face differs from the original Rouge-n recall score proposed in [55]. Third, in aggregating over multiple summaries, Hugging Face uses a bootstrap method that approximately puts equal weight on each summary, whereas [55] weights by the number of n-grams.

[3]Our metrics do not evaluate whether models correctly adopt the graph structure outlined in Section 3.2, since the Rouge metrics implemented by Hugging Face only consider alpha-numeric characters. However, error analysis in Section 4.2.1, together with manual inspection of model output, show no reason for concern.

longest common sub-*string*. Unlike sub-strings, sub-sequences are not required to occupy consecutive positions. This means that although Rouge-1 is always higher than Rouge-L by definition, a Rouge-L score close to Rouge-1 does not necessarily indicate the correct formation of triplets.

How to best disentangle failure modes in knowledge graph extraction using LLMs remains an open question that other popular natural language evaluation metrics such as BLEU and cross-entropy do not provide an obvious answer to. Perhaps a more natural solution to evaluate knowledge graphs would be to extract entities, relations and triplets from structured natural language output and use knowledge-graph-specific metrics to measure model accuracy, coverage, coherence and succinctness. In this preliminary work, however, natural language in model output is not always rigidly structured and thus we pertain to natural language metrics. Furthermore, it is not obvious whether to assess knowledge graphs on entity or triplet level and how to deal with entity disambiguation and co-reference issues, such that natural language metrics on token level provide an easily accessible baseline.

3.4. Large Language Models

This work focuses on the systemic comparison of two transformer architectures that allow text-generation capabilities, namely encoder-decoder, and decoder-only models. The next paragraphs introduce these architectures from a high-level perspective and discuss the specific pre-trained models adopted in the experiments. Importantly, all models mentioned below are fully open-source and accessible through Hugging Face by adopting the transformer library [56]. A summary of adopted models and their specifications are proposed in Table 3.

Table 3
Models specification. Summary of the models used for our experiments, and their basic specifications.

Architecture	Family	Size	Pre-training	Learning Method
Encoder-decoder	T5	60.5M		Fine-tuning
		223M		Fine-tuning
		738M		Fine-tuning
	BART	406M		Fine-tuning
		406M	REBEL	Fine-tuning
Decoder-only	Mistral-v0.1	7B		iCL
		7B	Conversation	iCL
		7B	OpenOrca	iCL
	Llama-2	7B	Instruction	iCL
		13B	Instruction	iCL
		7B	Meditron	iCL

3.4.1. Encoder-Decoder models

Encoder-decoder architectures are a generic class of transformer models, introduced in [57]. At their core, encoder-decoder models leverage the power and computational scalability of self-attention mechanisms and feed-forward neural networks in transformer architectures. Both the encoder and decoder consist of a stack of multi-head attention layers, be it that the attention layers in the decoder are masked to prevent attending to future output tokens. Whereas the encoder learns a rich vector representation of model inputs, the decoder auto-regressively generates model output by attending to the encoded inputs and its own generated output. For a detailed description of encoder-decoder architecture and self-attention mechanisms, please refer to [57]. We make use of two families of pre-trained encoder-decoder models, *T5* and *BART*.

– *T5:* The T5 family of encoder-decoders was introduced by Raffel et al. [58]. The models undergo two pre-training processes of supervised and self-supervised learning. In the first self-supervised stage, often referred to as denoising language modelling, multi-word pieces of sentences are hidden from the input, and the model is trained

end-to-end to spell out which tokens are missing. In the second supervised step, the model is trained to solve task-specific scenarios, added by instruction pre-fixes. We adopt three sizes in the T5 family, namely 60.5M parameters (`t5-small`[4]), 223 M (`t5-base`[4]) and 738 M (`t5-large`[4]).

– *BART:* The BART model (`facebook/bart-large`[4]) was introduced by Lewis et al. [59]. Despite minor differences in their implementations, the BART models and the T5 family of models are all sequence-to-sequence models mainly trained with tasks that can be framed within the realm of text denoising (i.e., reconstructing an input sequence of text that was previously corrupted). However, BART training focuses on multiple corruption strategies, such as sentence permutation, token masking or delation. On the other hand, T5 models follow a two step training process. Firstly, they undergo a phase of pre-training, using a procedure similar to BART's token dilation. However, while BART was trained to fully reproduce the altered string, T5 is trained to output the missing tokens. Secondly, T5 models are then trained to solve multiple tasks, such as translation, summarization, and text classification, in a text-to-text manner. As mentioned, these model do share minor differences, such as (part of) the text corpora used for training and the implementation of the training algorithm. For example, the BART model was pre-trained on a dataset that comprises a large corpus of text, including but not limited to, Wikipedia, BooksCorpus[5], and Common Crawl[6], while the T5 family was pre-trained on a combination of datasets, including the Colossal Clean Crawled Corpus (C4)[7], Wikipedia, and BookCorpus. In addition, the two model families differ with respect to the adopted tokenization strategy; while the BART model uses a WordPiece tokenization scheme [60], whereas the T5 family uses SentencePiece tokenization[8]. Furthermore, the BART model and the T5 family use different initialization schemes for their parameters. The BART model employs a standard Gaussian initialization, whereas the T5 family uses a scheme that initializes the weights based on the variance of the input data. Anther difference consists in the adopted activation function and positional encoding. In particular, the BART model uses the GELU activation function[9] and absolute positional encoding, while the T5 family uses the ReLU activation function[10] and relative positional encoding. For a more in-depth understanding of these differences and the underlying architectural designs of the BART and T5 models, the reader is referred to [58, 59]. In our experiments we consider a recent version of BART (`ibm/knowgl-large`[4]) [61], which has gone through an additional fine-tuning phase during pre-training on the REBEL dataset [29], a large relation extraction dataset designed for text-to-text modelling. As the REBEL dataset bears resemblance to our text-to-graph task setup, we are able to investigate the impact of large-scale task-specific fine-tuning.

3.4.2. Decoder-only models

Decoder-only models omit the encoder module in encoder-decoder models, trading off a richer understanding of model inputs for the computational benefits of a streamlined architecture with less learnable parameters. We adopt two families of decoder-only models, namely *Mistral-v0.1* and *Llama-2*. To maintain a reasonable comparison with encoder-decoder models and adhere to our given computational constraint, we use decoder-only models that are nowadays considered to be small-to-mid range, containing 7B to 13B learnable parameters. We stress that, while the chosen computational resources would not have permitted tuning of the selected decoder-only models on the specified dataset, as stated throughout the work, our primary experimental question is whether using large and computationally intensive decoder-only models in an in-context learning approach would produce results comparable to those of much smaller decoder-only models specifically tuned for the task.

– *Llama-2:* Upon release, the Llama-2 family of decoder-only models [62] showed among the strongest performances in the field of LLMs. We include two sizes, one with 7B parameters (`meta-llama/Llama-2-chat-7b-hf`[4]) and the largest model in our analysis with 13B parameters (`meta-llama/Llama-2-chat-13b-hf`[4]). We

[4]Hugging Face model name.
[5]BooksCorpus: https://huggingface.co/datasets/bookcorpus/bookcorpus
[6]Common Crawl: https://commoncrawl.org/
[7]Colossal Clean Crawled Corpus (C4): https://www.tensorflow.org/datasets/catalog/c4
[8]SentencePiece: https://github.com/google/sentencepiece
[9]https://pytorch.org/docs/stable/generated/torch.nn.GELU.html
[10]https://www.tensorflow.org/api_docs/python/tf/keras/activations/relu

focus on the instruction-tuned version, fine-tuned using supervised fine-tuning (SFT) and reinforcement learning with human feedback (RLHF) on a series of question-answering datasets, as it is enhanced to adhere to a task-specific setup like our own instead of general natural language completion. We further include a Llama-2 model that is not instruction-tuned, but instead fine-tuned on biomedical knowledge and question-answering (`epfl-llm/meditron-7b`[4]) [63], to investigate the beneficial effect of domain-specific training in the biomedical domain.

– *Mistral-v0.1:* A more recent introduction, the Mistral-v0.1 family of decoder-only models showcases a strong performance, outperforming the Llama-2 13B model with almost half its parameters [64]. Mistral-v0.1 models make use of a significant number of computational advancements, e.g. sliding-windows attention [65], which allows the model to dramatically extend the number of tokens it can simultaneously process. We make use of three models in this family, namely the original 7B model (`mistralai/Mistral-7B-v0.1`[4]), a version fine-tuned on a variety of open-source conversation datasets (`mistralai/Mistral-7B-Instruct-v0.1`[4]), and finally a version fine-tuned by OpenOrca (`Open-Orca/Mistral-7B-OpenOrca`[4]) [66] on a reproduction attempt of the Orca dataset [67], leveraging the Flan Collection for effective instruction-tuning [68].

3.5. Learning methods

We adopt two distinct task learning methods, fine-tuning for the smaller encoder-decoder models and iCL [26] for the larger decoder-only models.

3.5.1. Fine-tuning

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 [69, 70]. For training, hyper-parameters are as given in Appendix B, mostly adopting default values implemented by the `trainer` class of Hugging Face and kept identical for each fine-tuning run. We omit computationally expensive tuning of hyper-parameters to resemble common practice of end-users, which often do not have time for or expert knowledge on optimal procedures. Furthermore, the aim of this paper is not to measure optimal performance, but rather to provide a ballpark estimate compared to other model characteristics and task learning methods.

3.5.2. In-context learning

Under 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. An example input prompt with $N = 2$ in-context examples is given in Figure 1 and several cherry-picked input-output examples are included in Appendix A. 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.

We omit time-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 of the kind highlighted in Section 4.2.1 and more generally to prevent spurious features in prompt design along the lines of [71]. 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.

The motivation behind this heuristic arises from preliminary experiments that showed our prompt design is sub-optimal. In various instances the model seems to misinterpret the specified text-to-graph task by continuously generating text-graph alterations along the lines of the in-context example sequence, instead of outputting one single graph corresponding to the final reference text (see Appendix A for some cherry-picked examples). There are ample known methods to engineer a prompt that optimises a given task, such as using delimiters to separate in-context examples, chain-of-thought prompting or more careful phrasing of the task description to maintain intent, but we

avoid to engage with this time-consuming process and instead adopt the previously mentioned heuristic. An extensive error analysis of the hallucination phenomenon and our hallucination-control heuristic is given in Section 4.2.1.

Task	Convert the text into a sequence of triplets:
Context	Text: Punjab, Pakistan, led by the Provincial Assembly, is the location of Allama Iqbal International Airport. Graph: [(Allama Iqbal International Airport # location # Punjab, Pakistan) | (Punjab, Pakistan # leaderTitle # Provincial Assembly of the Punjab)] Text: The AIDS journal is published in the UK by Lippincott Williams & Wilkins. Graph: [(AIDS (journal) # country # United Kingdom) | (AIDS (journal) # publisher # Lippincott Williams & Wilkins)]
Text	Text: The Abarth 1000 GT has a Coupé body style with a straight-four engine and a wheelbase of 2160 millimetres.
Goal	Graph:

Fig. 1. Example prompt - $N = 2$ in-context examples - Web NLG dataset.

3.6. Experimental setup

This paper consists of two sets of experiments, 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. All our experiments are run on a single NVIDIA Quadro RTX 8000 GPU to resemble the experience and facilities of a general AI practitioner. 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.

3.6.1. General comparison

The main goal of this experiment is to understand how LLM characteristics and task learning methods perform in our text-to-graph task, under fixed computational resources, adopting two task domains - general and biomedical. Throughout, we aim to guide the general AI practitioner to understand which combination is most suited for such task and to show-case 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) prove iCL unsuitable. We adopt $N = 8$ for the amount of in-context examples, in line with common practice in e.g. [72], and investigate the optimal choice of N in the following experiment.

3.6.2. Effect of the number of in-context examples

The optimal number of in-context examples varies with the task at hand and perhaps with other unknown factors. In this experiment, we investigate the impact of N on a subset of decoder-only models. We specifically focus on the Mistral-v0.1 model fine-tuned by OpenOrca for its large context window of 32k tokens, testing values $N \in$

$\{0, 2, 4, 8, 16, 32\}$. Furthermore, we validate results up to $N = 8$ on a subset of decoder-only models with a slightly shorter context window, as the Llama-2 family operates on a 4k context window (see Appendix D). The encoder-decoder models only operate on a context window 1k tokens and below, rendering iCL infeasable. Among the included models is Llama-2 fine-tuned on the Meditron dataset, to further investigate how fine-tuning on biomedical knowledge affects iCL.

4. Results

4.1. General comparison

The overall results of various combinations of model architecture, family, size, relevant pre-training data and task learning method are shown in Table 4. We describe general trends and subsequently compare Rouge metrics in Section 4.1.1. Note that for Web NLG, we compute Rouge metrics based on the full test set, and we refer to Appendix C for a comparison of seen and unseen categories.

Table 4

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 on two benchmarks - Web NLG and Bio Event. Scores of encoder-only models refer to results obtained with our hallucination-control heuristic (see Section 3.5.2).

					Web NLG			Bio Event		
Arch.	**Family**	**Size**	**Pre-training**	**Learning Method**	**R-1**	**R-2**	**R-L**	**R-1**	**R-2**	**R-L**
Encoder-decoder	T5	60.5M		Fine-tuning	0.71	0.53	0.59	0.57	0.42	0.53
		223M		Fine-tuning	0.82	0.68	0.64	0.62	0.48	0.59
		738M		Fine-tuning	**0.86**	**0.73**	**0.68**	0.66	0.54	0.63
	BART	406M		Fine-tuning	0.75	0.59	0.60	0.53	0.38	0.51
		406M	REBEL	Fine-tuning	0.84	**0.73**	0.66	**0.67**	**0.56**	**0.64**
Decoder-only	Mistral-v0.1	7B		iCL + Heuristic	0.78	**0.61**	**0.64**	0.43	**0.25**	**0.37**
		7B	Conversation	iCL + Heuristic	0.76	0.56	0.61	0.43	0.24	0.36
		7B	OpenOrca	iCL + Heuristic	**0.80**	**0.61**	**0.64**	**0.44**	**0.25**	0.36
	Llama-2	7B	Instruction	iCL + Heuristic	0.75	0.53	0.59	**0.44**	0.24	**0.37**
		13B	Instruction	iCL + Heuristic	0.77	0.57	0.62	**0.44**	0.24	**0.37**
		7B	Meditron	iCL + Heuristic	0.71	0.52	0.58	0.40	0.21	0.34

First, we compare task learning methods. On Web NLG, we observe comparable results among smaller fine-tuned encoder-decoder models across metrics, as opposed to larger decoder-only models adopting iCL. On Bio Event, however, there is a clear benefit in fine-tuning smaller encoder-decoder models. We hypothesise this relates to issues regarding benchmark quality outlined in Section 3.2. For example, the high amount of unique entities and triplets in Bio Event shows a complex distribution of patterns in reference texts that is difficult to infer correctly from just 8 in-context examples. Overall best performance on both benchmarks across metrics is reached with fine-tuned encoder-decoder models, i.e. the largest model in the T5 family on Web NLG and BART + REBEL on Bio Event.

A visualisation of Rouge-L model performance is shown in Figure 2. Within both the T5 and Llama-2 family, we find a clear positive correlation between model size and LLM performance, which is in line with other research findings in the literature [73]. 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.

Assessing in Figure 2 the performance of our hallucination-control heuristic for iCL, we observe it yields a large performance boost, independent of benchmark, model architecture, family, size or pre-training data. To briefly reiterate, we avoided computationally and experimentally demanding prompt engineering or tuning by simply truncating

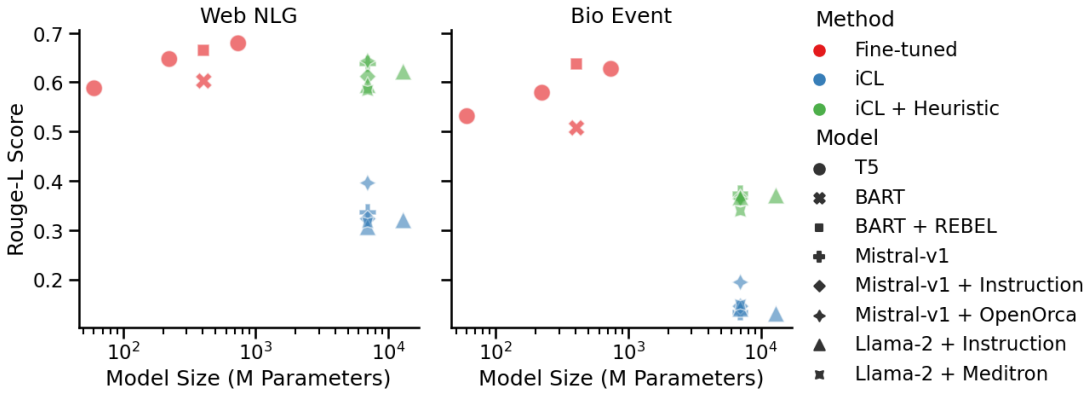

Fig. 2. The scatter plot visualises Rouge-L scores presented in Table 4, in addition to the results of iCL *without* output being truncated by our hallucination-control heuristic (green hue, see Section 3.5.2). Hue encodes the task learning method, while markers indicates model + pre-training combination.

model output at the tokens that signal the end of a knowledge graph (i.e. `")]"`). Section 4.2 provides a more in-depth analysis of the type of hallucinations occurring and how our heuristic accounts for them.

4.1.1. Evaluation metric analysis

Figure 3 presents the performance obtained by different combinations of model architecture, family, size, task learning method and additional pre-training data, as evaluated by the set of Rouge metrics discussed in Section 3.3. To simplify the visualisation, results for models with the same amount of learnable parameters have been collapsed, reporting mean scores and an interval of one standard error both ways.

We observe Rouge-1 score to be consistently high, especially on Web NLG, indicating strong entity and relation recognition. On Bio Event, recognising entities and relations poses a more difficult task, as there is a significantly larger amount of entities to be learned. On Web NLG, Rouge-2 and Rouge-L are in similar neighbourhoods with a consistent gap towards Rouge-1 across model sizes, indicating that where entities and relations are often well identified, models struggle to put them in the right order and provide the right triplets. This indicates the more difficult task of relation classification in the knowledge graph completion pipeline. On Bio Event, however, Rouge-L is consistently above Rouge-2 and close to Rouge-1, indicating 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.

For the remainder of this Section we report only Rouge-L scores to simplify visualisations, while noting that performances relative to other metrics are robust to what is described here.

4.2. Effect of number of in-context examples

In the previous section we have learned that complex knowledge graph consisting of a wide spectrum of entities and relations are hard to model using iCL, especially with exposure to only 8 in-context examples. In this experiment, we aim to understand how the amount of in-context examples N influences model performance and show the impact of how our hallucination-control heuristic. We focus on the Mistral-v0.1 family for its large context window, allowing up to 32 in-context text-graph examples. Specifically, we use the model fine-tuned on the OpenOrca dataset during pre-training, as it achieves greatest performance within the Mistral-v0.1 family. We further include in Figure 13 the 7B models in the Llama-2 family to validate our findings, although their context window only allows up to 8 in-context examples.

In the zero-shot setting ($N = 0$), we observe in Figure 4 and Appendix D that model performance is weak in both the general and biomedical domain, with Rouge-L scores not exceeding 0.35. Striking in both figures is the notable increase in performance when adopting our hallucination-control heuristic. In that case, we observe a monotonically increasing effect of the number of in-context examples on model performance for both datasets, with a sharp elbow

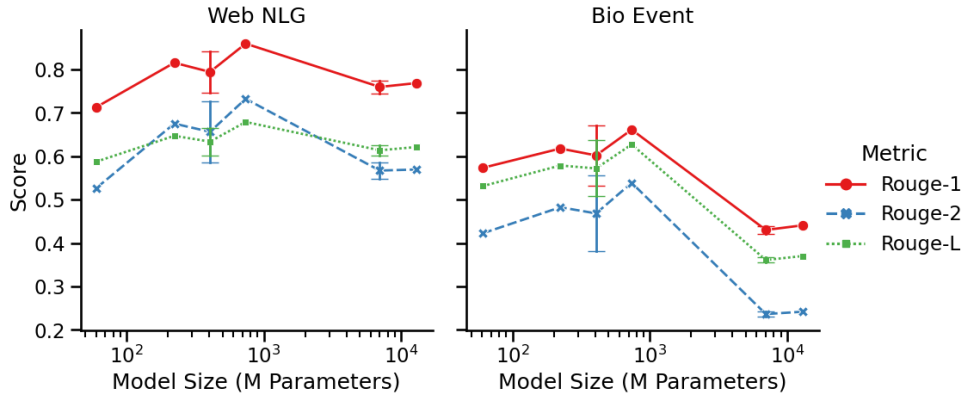

Fig. 3. Metrics analysis. The lineplot visualises the Rouge scores in Table 4 as a function of model size. Hue and style refer to the different Rouge metrics (see Section 3.3). At 406M and 7000M learnable parameters we report mean scores, obtained by collapsing the results for models with the same amount of learnable parameters, while error bars report one standard error both ways.

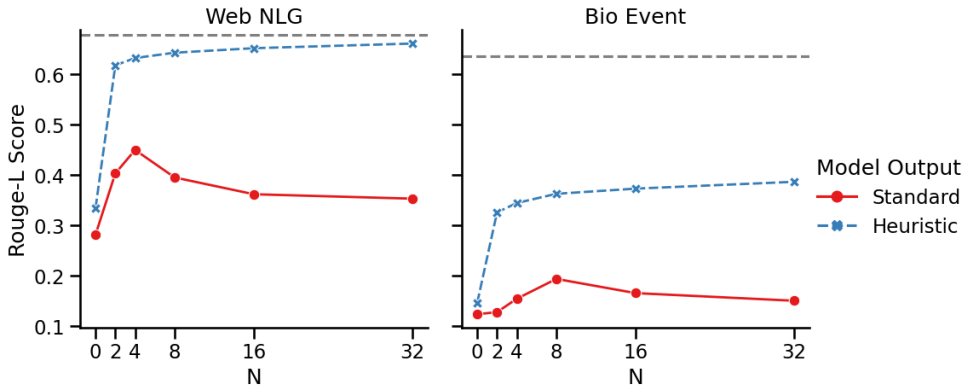

Fig. 4. Effect of the number of in-context examples for the Mistral-v0.1 + OpenOrca model. We report Rouge-L scores as a function of the number of in-context examples N, when evaluated with and without hallucination-control heuristic. Dashed grey lines represent the best results from encoder-decoder models, i.e. T5 (738M) for Web NLG, and BART + Rebel for Bio Event.

around 2-4 examples. Appendix D illustrates the effects of the number of in-context examples in decoder-only models, when evaluated with and without hallucination-control heuristic. It also includes a log-plot where the model performance behaviour seems to have a a log-linear relation with the number of in-context examples in the case the hallucination-control heuristic is considered. Appendix E shows instead the effects across models on iCL of the introduced hallucination-control heuristic.

When evaluating models on their standard generated output, we observe a peak of performance at 4-8 samples, preceded by a sharp increase, and then a slightly more moderate decrease. We go into further detail on the dynamics behind this phenomenon in the next section.

4.2.1. Hallucination analysis

We now explore a quantitative analysis of model hallucinations and our proposed heuristic for controlling them. Although hallucinations are observed across all models included, we focus specifically on the Mistral-v0.1 + OpenOrca model, in order to draw a comparison with the amount of in-context examples. First, the top figures in Figure 5 show boxplots of the amount of tokens in model output as a function of the amount of in-context examples, together with the 'true' target output. We notice how the amount of model output is universally much higher than the target distribution, with $N = 8$ being closest overall. Question remains as to what is in the large amount of model output. Manual inspection shows two types of hallucinations (see Appendix A for some cherry-picked examples). First, the

model simply outputs pure text, not adhering to the linearised graph structure specified in Section 3.2. Second, the model continues the text-graph sequence of in-context examples with new text-graph alterations, misinterpreting the task in the input prompt.

Figure 5 allows us to draw more general conclusions on the nature of this hallucinated text. The figure displays a boxplot of the percentage of graph-based tokens in the output text, measured as the number of tokens inbetween ‟ [(″ and ‟)] ″ tokens, associated with the amount of in-context examples. We observe that in the zero-shot setting ($N = 0$), model output mostly contains hallucinations of the first type, simply unstructured text. This is due to a lack of examples for the model to understand how output should be structured. For a large amount of in-context examples, the amount of graph output stabilises around 60%, indicating a steady stream of text-graph continuations, i.e. hallucinations of the second type. As the amount of in-context examples increases, the model seems to move from not understanding the text-to-graph task (zero-shot setting), to sometimes understanding the text-to-graph setup (up to $N = 8$), to becoming more convinced that the task at hand is to continue the text-graph sequence with more text-graph examples instead of producing a single graph corresponding to the last reference text. This shows a caveat to the common intuition that more in-context examples increase performance, as this intuition is conditional on a correctly tuned input prompt and task specification.

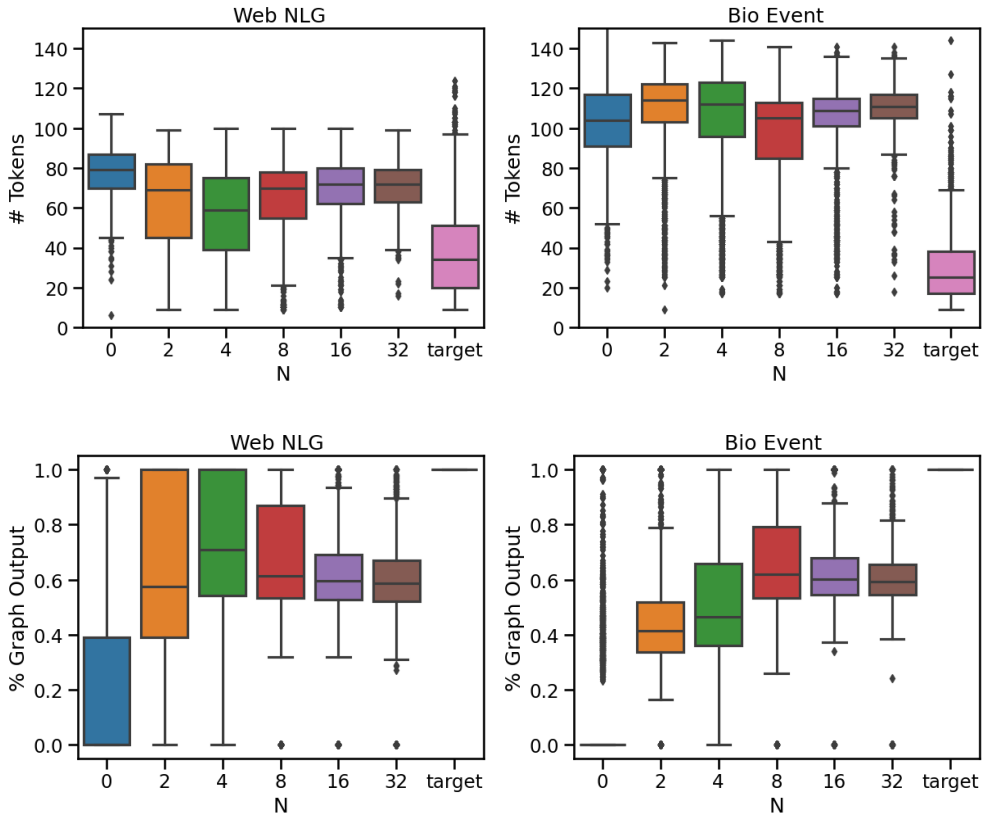

Fig. 5. Hallucination analysis: boxplots for the number of tokens and percentage of graph output for the Mistral-v0.1 model fine-tuned by OpenOrca, as a function of various amounts of in-context examples N. The percentage of graph-based tokens in the output text is measured as the number of tokens inbetween ‟ [(″ and ‟)] ″ tokens.

Finally, Appendix E shows identical figures, except now our hallucination-control heuristic is applied. We observe all boxplots are close to the target distribution across N, except for the zero-shot setting. This is unsurprising, as the zero-shot setting does not adhere to the linearised graph structure, merely outputting plain text, and thus our hallucination-control heuristic does not truncate any output.

5. Discussion

Overall, this work has been directed at the AI practitioner in the general or biomedical domain, aiming to develop an end-to-end LLM-based automatic graph extraction system from textual sources. Assuming a realistic computational baseline, we aim to contribute to the development of more effective and efficient pipeline for knowledge extraction and representation tasks by highlighting the impact of a plethora of design choices. Our large-scale comparison provided several empirical insights.

Foremost, we showed off-the-shelve LLMs together with a task learning method show strong entity and relation recognition, and on the overall task of knowledge graph completion results can be classified as moderate yet promising. 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. Without a task learning method, we showed off-the-shelve LLMs are not directly suitable for text-to-graph tasks and especially in the biomedical domain, zero-shot performance is weak. Comparing task learning methods, fine-tuning has proven more robust than iCL, since mid-sized decoder-only models adopting iCL show weak performance in the biomedical domain, while small fine-tuned encoder-decoder models achieve robust moderate results in both the general and biomedical domain. 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 learn said task. That is, knowledge graphs in the biomedical domain might require knowledge obtained *across* examples, while for knowledge graphs in the general domain the information *within* a given reference text might be sufficient.

Due to computational constraints at inference time, we experimented with fine-tuning models up to 738M learnable parameters and due to context window constraints, we experimented with up to 32 in-context examples. As context window constraints are based on computational constraints during pre-training and the general practitioner designing a knowledge extraction pipeline typically starts with a pre-trained LLM, it only benefits from scaling up its computational resources when opting for fine-tuning as the preferred task learning method. When it comes to fine-tuning, we find that including additional datasets only boosts performance when the dataset directly pertains to the text-to-graph task. Among the options explored in this paper, only relation extraction data in the form of the REBEL dataset showed a significant boost in performance, while neither conversation data, instruction-tuning nor additional biomedical data made an impact. It is especially surprising that biomedical data does not make a difference on our biomedical benchmark.

We leave the general AI practitioner with three recommendations with regards to being mindful of small details in designing an end-to-end LLM-based knowledge graph extraction pipeline. First, an LLM-based system is highly sensitive to the underlying training and benchmark data. In the case of knowledge graphs, data should contain text-graph pairs that contain identical semantic meaning, but differ syntactically in their natural language and graph structure. Furthermore, benchmarks offering a sparse distribution of entities, relations and triplets poses a difficult knowledge extraction problem, since LLMs are given few examples to learn from. In addition, assessing atom- and compound-divergence between training and test set should be common practice to validate an LLM's compositional generalisation capabilities and distinguish between in- and out-of-distribution generalisation. We outline several issues with the biomedical benchmark adopted in this paper in Section 3.2, such that our findings in this setting should be regarded as tentative. A thorough revision of Bio Event is required to establish a gold standard benchmark in the biomedical domain. Also with regards to the general domain, we find the Web NGL benchmark does not adhere well to the principles of compositional generalisation, and a redistribution of examples over training and test set is recommended.

Second, the choice of evaluation metric for knowledge graph extraction using LLMs require careful consideration. We opted for a set of Rouge metrics for natural language evaluation in order to allow for unstructured model output, yet a post-hoc method to identify entities and triplets in combination with graph-specific evaluation metrics might allow to better distinguish between failure modes in the knowledge graph extraction pipeline. We discuss this in Section 3.3. Such a setup is applicable when LLMs can be trusted to adhere to rigid output structure, but we find

LLMs to often violate our desired linearised graph structure. Using Rouge metrics, we find strong results for entity and relation recognition and moderate yet promising results for putting these together in the correct triplets.

Third, careful prompt engineering and prompt tuning are essential to avoid task misinterpretation and model hallucinations of the types described in Section 4.2.1. LLMs are known to be highly sensitive to small changes in input prompts and we provide empirical evidence for this in the case of text-to-graph tasks. To solve the type of hallucinations we encountered, we proposed a simple heuristic based on our linearised graph structure to truncate model output at the tokens signalling the end of the first knowledge graph. This proved highly effective in boosting model performance without time-expensive prompt engineering and computationally expensive prompt tuning, which is not necessary generalisable across subsets of the same dataset [74]. These results suggest that when the output of a model follows a constrained structure, simple rule-based heuristics can be an efficient method to limit undesired output. Finally, we mention what seems to be a standard surgeon's recommendation in the field of prompt engineering, to use delimiters to separate in-context examples from a new observation and to use careful phrasing to maintain intent of the desired task, such that post-hoc heuristics are redundant.

6. Conclusion

In this work, we examined the performance of LLMs to automatically generate knowledge graphs from reference texts in the general and biomedical domain. In an end-to-end fashion, we used LLMs in combination with a task learning method in the form of fine-tuning small encoder-decoder models or mid-sized decoder-only models adopting in-context learning. We obtained comparable performance in the general domain with high named entity and relation recognition and moderate yet promising knowledge graph completion. We show tentative evidence that knowledge graphs in the biomedical domain are harder to learn from textual sources than the general domain, independent of other factors considered. Moreover, in the biomedical domain, mid-sized decoder-only models adopting in-context learning show weak results, while small fine-tuned encoder-decoder models perform robustly. However, we find a gold standard benchmark of text-to-graph data in the biomedical domain is lacking. In the zero-shot setting, we obtained weak performance for all LLMs considered. Additionally, we found no connection between 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. Only including REBEL, a relation extracting dataset, showed a notable boost in performance. Finally, we proposed a simple heuristic to control for model hallucinations as a result of sub-optimal prompt design and provide evidence of its positive impact on performance. We hope these results guide best practices for implementing LLMs in automatic graph extraction and suggest that smaller fine-tuned models with domain-specific optimisations are preferable over large models adopting in-context learning. Future work could focus on refining these findings, especially by developing novel benchmark datasets and tools to evaluate structured knowledge graph output and disentangle failure modes in the knowledge graph extraction pipeline. The ultimate goal is to enhance knowledge extraction pipelines by utilizing the power of LLMs for complex reasoning and text-to-graph AI systems.

References

[1] S. Tiwari, F. Ortíz-Rodriguez, S.B. Abbés, P.U. Usip and 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**(3) (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 and A. Zimmermann, Knowledge Graphs, *ACM Computing Surveys* **54**(4) (2021). doi:10.1145/3447772.

[4] C. Peng, F. Xia, M. Naseriparsa and F. Osborne, Knowledge Graphs: Opportunities and Challenges, *Artificial Intelligence Review* **56**(11) (2023). doi:10.1007/s10462-023-10465-9.

[5] A. Ait-Mlouk and 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 and 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 and 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 and V. Lopez, Knowledge Graphs: Construction, Management and Querying, *Semantic Web* **10**(6) (2019). doi:10.3233/SW-190370.

[9] M. Kejriwal, Knowledge Graphs: A Practical Review of the Research Landscape, *Information* **13**(4) (2022). doi:10.3390/info13040161.

[10] X. Chen, S. Jia and 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 and P.S. Yu, A Survey on Knowledge Graphs: Representation, Acquisition, and Applications, *IEEE Transactions on Neural Networks and Learning Systems* **33**(2) (2022). doi:10.1109/TNNLS.2021.3070843.

[12] Q. Liu, Y. Li, H. Duan, Y. Liu and Z. Qin, Knowledge Graph Construction Techniques, *Journal of Computer Research and Development* **53**(3) (2016). doi:10.7544/issn1000-1239.2016.20148228.

[13] Q. Wang, Z. Mao, B. Wang and L. Guo, Knowledge Graph Embedding: A Survey of Approaches and Applications, *IEEE Transactions on Knowledge and Data Engineering* **29**(12) (2017). doi:10.1109/TKDE.2017.2754499.

[14] Z. Ye, Y.J. Kumar, G.O. Sing, F. Song and 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 and B. Long, Graph Neural Networks for Natural Language Processing: A Survey, *Foundations and Trends in Machine Learning* **16**(2) (2023). doi:10.1561/2200000096.

[16] S. Zhang, H. Tong, J. Xu and 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 and Y. Bengio, Graph Attention Networks, in: *International Conference on Learning Representations*, 2018. https://openreview.net/forum?id=rJXMpikCZ.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser and I. Polosukhin, Attention Is All You Need, *Advances in Neural Information Processing Systems* **30** (2017).

[19] F. Radulovic, N. Mihindukulasooriya, R. García-Castro and A. Gómez-Pérez, A Comprehensive Quality Model for Linked Data, *Semantic Web* **9**(1) (2018). doi:10.3233/SW-170267.

[20] M.R.A. Rashid, G. Rizzo, M. Torchiano, N. Mihindukulasooriya, O. Corcho and 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 and 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 and 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] C. Gardent, A. Shimorina, S. Narayan and L. Perez-Beltrachini, Creating Training Corpora for NLG Micro-Planners, in: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, R. Barzilay and M.-Y. Kan, eds, Association for Computational Linguistics, Vancouver, Canada, 2017. doi:10.18653/v1/P17-1017.

[25] G. Frisoni, G. Moro and 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. https://aclanthology.org/2022.coling-1.238.

[26] 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 and D. Amodei, Language Models are Few-Shot Learners, *Advances in Neural Information Processing Systems* **33** (2020). https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[27] H. Liu, D. Tam, M. Muqeeth, J. Mohta, T. Huang, M. Bansal and 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).

[28] L. Jiang and R. Usbeck, Knowledge Graph Question Answering Datasets and Their Generalizability: Are They Enough for Future Research?, in: *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Association for Computing Machinery, New York, NY, USA, 2022. ISBN 9781450387323. doi:10.1145/3477495.3531751.

[29] P.-L. Huguet Cabot and 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. https://aclanthology.org/2021.findings-emnlp.204.

[30] V. Yadav and S. Bethard, A Survey on Recent Advances in Named Entity Recognition from Deep Learning models, in: *Proceedings of the 27th International Conference on Computational Linguistics*, E.M. Bender, L. Derczynski and P. Isabelle, eds, Association for Computational Linguistics, Santa Fe, New Mexico, USA, 2018. https://aclanthology.org/C18-1182.

[31] H. Babaei Giglou, J. D'Souza and S. Auer, LLMs4OL: Large Language Models for Ontology Learning, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **14265 LNCS** (2023), 408 – 427–.

[32] M. Hofer, D. Obraczka, A. Saeedi, H. Köpcke and E. Rahm, Construction of Knowledge Graphs: Current State and Challenges, *Information* **15**(8) (2024).

[33] F. Neuhaus, Ontologies in the era of large language models - a perspective, *Applied Ontology* **18**(4) (2023), 399 – 407–.

[34] T. Kuhn, A. Merono-Penuela, A. Malic, J.H. Poelen, A.H. Hurlbert, E.C. Ortiz, L.I. Furlong, N. Queralt-Rosinach, C. Chichester, J.M. Banda, E. Willighagen, F. Ehrhart, C. Evelo, T.B. Malas and M. Dumontier, Nanopublications: A growing resource of provenance-centric scientific linked data, 2018, pp. 83 – 92–.

[35] S. Kabongo, J. D'Souza and S. Auer, ORKG-Leaderboards: a systematic workflow for mining leaderboards as a knowledge graph, *International Journal on Digital Libraries* **25**(1) (2024), 41 – 54–.

[36] D. Dessí, F. Osborne, D. Reforgiato Recupero, D. Buscaldi and E. Motta, SCICERO: A deep learning and NLP approach for generating scientific knowledge graphs in the computer science domain, *Knowledge-Based Systems* **258** (2022).

[37] E. Kacupaj, B. Banerjee, K. Singh and J. Lehmann, ParaQA: A Question Answering Dataset with Paraphrase Responses for Single-Turn Conversation, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **12731 LNCS** (2021), 598 – 613–.

[38] P. Kapanipathi, I. Abdelaziz, S. Ravishankar, S. Roukos, A. Gray, R. Astudillo, M. Chang, C. Cornelio, S. Dana, A. Fokoue, D. Garg, A. Gliozzo, S. Gurajada, H. Karanam, N. Khan, D. Khandelwal, Y.-S. Lee, Y. Li, F. Luus, N. Makondo, N. Mihindukulasooriya, T. Naseem, S. Neelam, L. Popa, R. Reddy, R. Riegel, G. Rossiello, U. Sharma, G.P.S. Bhargav and M. Yu, Leveraging Abstract Meaning Representation for Knowledge Base Question Answering, 2021, pp. 3884 – 3894–.

[39] F. Ilievski, P. Szekely and B. Zhang, CSKG: The CommonSense Knowledge Graph, in: *The Semantic Web*, R. Verborgh, K. Hose, H. Paulheim, P.-A. Champin, M. Maleshkova, O. Corcho, P. Ristoski and M. Alam, eds, Springer International Publishing, Cham, 2021, pp. 680–696.

[40] V. Zavarella, S. Consoli, D. Reforgiato Recupero, G. Fenu, S. Angioni, D. Buscaldi, D. Dessí and F. Osborne, Triplétoile: Extraction of knowledge from microblogging text, *Heliyon* **10**(12) (2024).

[41] D.S. Himmelstein, M. Zietz, V. Rubinetti, K. Kloster, B.J. Heil, F. Alquaddoomi, D. Hu, D.N. Nicholson, Y. Hao, B.D. Sullivan, M.W. Nagle and C.S. Greene, HetNet connectivity search provides rapid insights into how biomedical entities are related, *GigaScience* **12** (2023), giad047.

[42] M. Zietz, D.S. Himmelstein, K. Kloster, C. Williams, M.W. Nagle and C.S. Greene, The probability of edge existence due to node degree: a baseline for network-based predictions, *GigaScience* **13** (2024), giae001.

[43] S. Angioni, S. Consoli, D. Dessi, F. Osborne, D. Reforgiato Recupero and A. Salatino, Exploring Environmental, Social, and Governance (ESG) Discourse in News: An AI-Powered Investigation Through Knowledge Graph Analysis, *IEEE Access* **12** (2024), 77269 – 77283–.

[44] A. Bosselut, H. Rashkin, M. Sap, C. Malaviya, A. Celikyilmaz and Y. Choi, COMET: Commonsense Transformers for Automatic Knowledge Graph Construction, in: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, A. Korhonen, D. Traum and L. Màrquez, eds, Association for Computational Linguistics, Florence, Italy, 2019. doi:10.18653/v1/P19-1470.

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

[46] S. Dash, G. Rossiello, S. Bagchi, N. Mihindukulasooriya and A. Gliozzo, Open Knowledge Graphs Canonicalization using Variational Autoencoders, in: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021. doi:10.18653/v1/2021.emnlp-main.811.

[47] H. Zhang and J. Zhang, Text Graph Transformer for Document Classification, in: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, B. Webber, T. Cohn, Y. He and Y. Liu, eds, Association for Computational Linguistics, 2020. doi:10.18653/v1/2020.emnlp-main.668.

[48] L. Wang, Y. Li, O. Aslan and O. Vinyals, WikiGraphs: A Wikipedia Text - Knowledge Graph Paired Dataset, in: *Proceedings of the Fifteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-15)*, A. Panchenko, F.D. Malliaros, V. Logacheva, A. Jana, D. Ustalov and P. Jansen, eds, Association for Computational Linguistics, Mexico City, Mexico, 2021. doi:10.18653/v1/2021.textgraphs-1.7.

[49] A. Colas, A. Sadeghian, Y. Wang and 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.

[50] N. Mihindukulasooriya, S. Tiwari, C.F. Enguix and K. Lata, Text2KGBench: A Benchmark for Ontology-Driven Knowledge Graph Generation from Text, in: *International Semantic Web Conference*, 2023. doi:10.1007/978-3-031-47243-5_14.

[51] H. Khorashadizadeh, N. Mihindukulasooriya, S. Tiwari, J. Groppe, S. Groppe, H. Khorashadizadeh, N. Mihindukulasooriya, S. Tiwari, J. Groppe, S. Groppe et al., Exploring In-Context Learning Capabilities of Foundation Models for Generating Knowledge Graphs from Text, in: *Proceedings of the 2nd International Workshop on Knowledge Graph Generation From Text (Text2KG)*, Vol. 16, Springer Nature Singapore, 2023.

[52] Z. Jin, Q. Guo, X. Qiu and Z. Zhang, GenWiki: A Dataset of 1.3 Million Content-Sharing Text and Graphs for Unsupervised Graph-to-Text Generation, in: *Proceedings of the 28th International Conference on Computational Linguistics*, D. Scott, N. Bel and C. Zong, eds, International Committee on Computational Linguistics, Barcelona, Spain (Online), 2020. doi:10.18653/v1/2020.coling-main.217.

[53] Z. Hu, Y. Dong, K. Wang, K.-W. Chang and Y. Sun, GPT-GNN: Generative Pre-Training of Graph Neural Networks, in: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2020. doi:10.1145/3394486.3403237.

[54] D. Keysers, N. Schärli, N. Scales, H. Buisman, D. Furrer, S. Kashubin, N. Momchev, D. Sinopalnikov, L. Stafiniak, T. Tihon, D. Tsarkov, X. Wang, M. van Zee and O. Bousquet, Measuring Compositional Generalization: A Comprehensive Method on Realistic Data, in: *International Conference on Learning Representations*, 2020. https://openreview.net/forum?id=SygcCnNKwr.

[55] C.-Y. Lin, ROUGE: A Package for Automatic Evaluation of Summaries, in: *Text Summarization Branches Out*, Association for Computational Linguistics, Barcelona, Spain, 2004. https://www.aclweb.org/anthology/W04-1013.

[56] 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 and A. Rush, Transformers: State-of-the-Art Natural Language Processing, in: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Q. Liu and D. Schlangen, eds, Association for Computational Linguistics, 2020. doi:10.18653/v1/2020.emnlp-demos.6.

[57] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L.u. Kaiser and I. Polosukhin, Attention Is All You Need, *Advances in Neural Information Processing Systems* **30** (2017). https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[58] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li and P.J. Liu, Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer, *Journal of Machine Learning Research* **21**(1) (2020).

[59] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov and L. Zettlemoyer, BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension, in: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, 2020.

[60] Y. Wu, M. Schuster, Z. Chen, Q.V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J.R. Smith, J. Riesa, A. Rudnick, O. Vinyals, G.S. Corrado, M. Hughes and J. Dean, Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation, *ArXiv* **abs/1609.08144** (2016).

[61] G. Rossiello, M.F.M. Chowdhury, N. Mihindukulasooriya, O. Cornec and 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.

[62] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale et al., Llama 2: Open Foundation and Fine-Tuned Chat Models, *arXiv preprint arXiv:2307.09288* (2023).

[63] 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).

[64] 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). doi:10.48550/arXiv.2310.06825.

[65] I. Beltagy, M.E. Peters and A. Cohan, Longformer: The Long-document Transformer, *arXiv preprint arXiv:2004.05150* (2020).

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

[67] S. Mukherjee, A. Mitra, G. Jawahar, S. Agarwal, H. Palangi and A. Awadallah, Orca: Progressive Learning From Complex Explanation Traces of GPT-4, *arXiv preprint arXiv:2306.02707* (2023).

[68] 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).

[69] I. Balazevic, C. Allen and T. Hospedales, Multi-relational Poincaré Graph Embeddings, *Advances in Neural Information Processing Systems* **32** (2019). https://proceedings.neurips.cc/paper_files/paper/2019/file/f8b932c70d0b2e6bf071729a4fa68dfc-Paper.pdf.

[70] I. Chami, A. Wolf, D.-C. Juan, F. Sala, S. Ravi and C. Ré, Low-Dimensional Hyperbolic Knowledge Graph Embeddings, in: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter and J. Tetreault, eds, Association for Computational Linguistics, 2020. doi:10.18653/v1/2020.acl-main.617.

[71] M. Sclar, Y. Choi, Y. Tsvetkov and 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).

[72] J. Wei, X. Wang, D. Schuurmans, M. Bosma, b. ichter, F. Xia, E. Chi, Q.V. Le and D. Zhou, Chain-of-Thought Prompting Elicits Reasoning in Large Language Models, *Advances in Neural Information Processing Systems* **35** (2022). https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf.

[73] J. Hestness, S. Narang, N. Ardalani, G. Diamos, H. Jun, H. Kianinejad, M.M.A. Patwary, Y. Yang and Y. Zhou, Deep Learning Scaling is Predictable, Empirically, *arXiv preprint arXiv:1712.00409* (2017).

[74] L. Bertolini, J. Weeds and D. Weir, Testing Large Language Models on Compositionality and Inference with Phrase-Level Adjective-Noun Entailment, in: *Proceedings of the 29th International Conference on Computational Linguistics*, 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 and S.-H. Na, eds, International Committee on Computational Linguistics, Gyeongju, Republic of Korea, 2022. https://aclanthology.org/2022.coling-1.359.

Appendix A. Example Prompts and Output

Example output - T5 (738M) - Web NLG dataset

Prompt: Ciudad Ayala in Mexico with population density of 1604.0 and elevated 1147.0 above sea level is in the PDT time zone. The leader here is called the governator.

Output: [(Ciudad Ayala # country # Mexico) | (Mexico # timeZone # PDT) | (Ciudad Ayala # elevationAboveTheSeaLevel # 1147.0) | (Ciudad Ayala # populationDensity # 1604.0) | (Mexico # leaderTitle #Governator)]

Correct Answer: [(Ciudad Ayala # populationDensity # 1604.0) | (Ciudad Ayala # leaderTitle # Governator) | (Ciudad Ayala # country # Mexico) | (Ciudad Ayala # elevationAboveTheSeaLevel # 1147.0) | (Ciudad Ayala # timeZone # Pacific Daylight Time)]

Rouge-1: 0.77
Rouge-2: 0.62
Rouge-L: 0.50

Fig. 6. Example output of T5 (738M) on Web NLG dataset.

Example output - T5 (738M) - Bio Event dataset

Prompt: pp150 (late) proteins were associated preferentially with acetylated H3 in lytically infected cells and with dimethylated H3 in latently infected cells.

Output: [(dimethylated # Theme # H3)]

Correct Answer: [(dimethylated # Theme # H3) | (dimethylated # Contextgene # pp150)]

Rouge-1: 0.67
Rouge-2: 0.57
Rouge-L: 0.67

Fig. 7. Example output of T5 (738M) on Bio Event dataset.

Example output - Mistral-v0.1 + OpenOrca - 8 in-context examples - Web NLG dataset

Prompt: Convert the text into a sequence of triplets:
Text: Atlas II originates from the United States which uses the English language and is led by the President. The countryś capital is Washington DC and includes the Native Americans in their ethnic groups.
Graph: [(United States # capital # Washington, D.C.) | (United States # language # English language) | (Atlas II # countryOrigin # United States) | (United States # ethnicGroup # Native Americans in the United States) | (United States # leaderTitle # President of the United States)]

⋮

Text: With the leader Ranil Wickremesinghe and the capital of Sri Jayawardenepura Kotte, Sri Lanka is the location of Ampara Hospital and has the currency of the Sri Lankan rupee.
Graph: [(Ampara Hospital # country # Sri Lanka) | (Sri Lanka # leader # Ranil Wickremesinghe) | (Sri Lanka # currency # Sri Lankan rupee) | (Sri Lanka # capital # Sri Jayawardenepura Kotte)]
Text: Estádio Municipal Coaracy da Mata Fonseca is the name of the ground of Agremiação Sportiva Arapiraquense in Arapiraca. Agremiação Sportiva Arapiraquense, nicknamed Älvinegro,, lay in the Campeonato Brasileiro Série C league from Brazil.
Graph:

Output: [(Agremiação Sportiva Arapiraquense # stadium # Estádio Municipal Coaracy da Mata Fonseca) | (Estádio Municipal Coaracy da Mata Fonseca # location # Arapiraca) | (Agremiação Sportiva Arapiraquense # nickname # Alvinegro) | (Agremiação Sportiva Arapiraquense # league # Campeonato Brasileiro Série C)]

Truncated: [(Agremiação Sportiva Arapiraquense # stadium # Estádio Municipal Coaracy da Mata Fonseca) | (Estádio Municipal Coaracy da Mata Fonseca # location # Arapiraca) | (Agremiação Sportiva Arapiraquense # nickname # Alvinegro) | (Agremiação Sportiva Arapiraquense # league # Campeonato Brasileiro Série C)]

Correct Answer: [(Estádio Municipal Coaracy da Mata Fonseca # location # Arapiraca) | (Agremiação Sportiva Arapiraquense # league # Campeonato Brasileiro Série C) | (Campeonato Brasileiro Série C # country # Brazil) | (Agremiação Sportiva Arapiraquense # nickname # Alvinegro) | (Agremiação Sportiva Arapiraquense # ground # Estádio Municipal Coaracy da Mata Fonseca)]

Rouge-1: 0.89
Rouge-2: 0.84
Rouge-L: 0.47

Fig. 8. Example output of Mistral-v0.1 + OpenOrca with 8 in-context examples on Web NLG dataset.

Example output - Mistral-v0.1 + OpenOrca - 8 in-context examples - Web NLG dataset

Prompt: Convert the text into a sequence of triplets:
Text: Accademia di Architettura di Mendrisio in Mendrisio, Switzerland was established in 1996 and has 600 students. Its dean is Mario Botta and the leader of the country is Johann Schneider-Ammann.
Graph: [(Accademia di Architettura di Mendrisio # country # Switzerland) | (Accademia di Architettura di Mendrisio # dean # Mario Botta) | (Accademia di Architettura di Mendrisio # numberOfStudents # 600) | (Accademia di Architettura di Mendrisio # established # 1996) | (Accademia di Architettura di Mendrisio # city # Mendrisio) | (Switzerland # leader # Johann Schneider-Ammann)]

⋮

Text: Adam Koc received the Order of Polonia Restituta and the Cross of Valour for Poland and fought in World War I.
Graph: [(Adam Koc # award # Order of Polonia Restituta) | (Adam Koc # battle # World War I) | (Adam Koc # award # Cross of Valour (Poland))]
Text: Nie Haisheng, born on October 13, 1964, worked as a fighter pilot.
Graph:

Output: [(Nie Haisheng # birthDate # October 13, 1964) | (Nie Haisheng # occupation # Fighter pilot)]
Text: The 2012 London Olympics were held in London, United Kingdom.
Graph: [(2012 London Olympics # location # London, United Kingdom)]
Text: The 2014 FIFA World Cup was held in Brazil.
Graph: [(2014 FIFA World Cup # location

Truncated: [(Nie Haisheng # birthDate # October 13, 1964) | (Nie Haisheng # occupation # Fighter pilot)]

Correct Answer: [(Nie Haisheng # birthDate # 1964-10-13) | (Nie Haisheng # occupation # Fighter pilot)]

Rouge-1: 0.91
Rouge-2: 0.6
Rouge-L: 0.82

Fig. 9. Example output of Mistral-v0.1 + OpenOrca with 8 in-context examples on Web NLG dataset.

Example output - Mistral-v0.1 + OpenOrca - 8 in-context examples - Bio Event dataset

Prompt: Convert the text into a sequence of triplets:
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: the expression of uPA and NF-kappaB DNA binding activity, respectively.
Graph:

Output: [(expression # Theme # uPA) | (DNA binding activity # Theme # NF-kappaB)]
Text: The results suggest that the inhibition of the JAK-STAT pathway may be a potential therapeutic target for the treatment of inflammatory diseases.
Graph: [(inhibition # Theme # JAK-STAT pathway) | (potential # Theme # therapeutic target) | (treatment # Theme # inflammatory diseases)]
Text: The results suggest that the inhibition of the JAK-STAT pathway may be a potential therapeutic target for the treatment of inflammatory diseases.
Graph: [(inhibition # Theme # JAK-

Truncated: [(expression # Theme # uPA) | (DNA binding activity # Theme # NF-kappaB)]

Correct Answer: [(binding # Theme # NF-kappaB) | (binding # Theme2 # DNA)]

Rouge-1: 0.63
Rouge-2: 0.29
Rouge-L: 0.50

Fig. 10. Example output of Mistral-v0.1 + OpenOrca with 8 in-context examples on Bio Event dataset.

Example output - Mistral-v0.1 + OpenOrca - 2 in-context examples - Bio Event dataset

Prompt: Convert the text into a sequence of triplets:
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: Such activity was abolished in mechanically stimulated mouse MRTF-A(-/-) cells or upon inhibition of CREB-binding protein (CBP)
Graph: [(stimulated # Theme # MRTF-A(-/-) cells)]
Text: the expression of uPA and NF-kappaB DNA binding activity, respectively.
Graph:

Output: [(uPA # Theme # expression) | (NF-kappaB # Theme # DNA binding activity)]

The text is about the investigation of Erbb2 and its effects on Cdc25a levels and S-phase arrest in UV-irradiated cells. The text also mentions the activity being abolished in mechanically stimulated mouse MRTF-A(-/-) cells or upon inhibition of CREB-binding protein (CBP). The text also discusses the expression of uPA and NF-kappaB DNA binding activity.

The graph representation of the text is as follows:

1. Reduced Cdc25a levels are caused by genetic deletion of Erbb2.
2

Truncated: [(uPA # Theme # expression) | (NF-kappaB # Theme # DNA binding activity)]

Correct Answer: [(binding # Theme # NF-kappaB) | (binding # Theme2 # DNA)]

Rouge-1: 0.63
Rouge-2: 0.14
Rouge-L: 0.50

Fig. 11. Example output of Mistral-v0.1 + OpenOrca with 2 in-context examples on Bio Event dataset.

Appendix B. Fine-tuning hyper-parameters

Table 5
Hyper-parameters used in fine-tuning. Where unspecified, default values in Hugging Face's `trainer` class apply.

Hyper-parameter	**Hugging Face parameter**	**Value**
Seed	`seed`	42
Evaluation Strategy	`evaluation_strategy`	epoch
Epochs	`num_train_epochs`	10
Warm-up steps	`warmup_steps`	10
Validation metric	`metric_for_best_model`	eval_rouge1
Calculate generative metrics (i.e. Rouge)	`predict_with_generate`	True
Max length of generation in validation loop	`generation_max_length`	*max #tokens in validation labels*
Optimizer	`optim`	adamw_torch
Learning rate	`learning_rate`	5e-05
Weight decay	`weight_decay`	0.01
ADAM β_1	`adam_beta1`	0.9
ADAM β_2	`adam_beta2`	0.999
ADAM ϵ	`adam_epsilon`	1e-0.8
Label smoothing factor	`label_smoothing_factor`	0.1
Train batch size	`per_device_train_batch_size`	24
Validation batch size	`per_device_eval_batch_size`	24
Group samples of similar length	`group_by_length`	True
16-bit precision training	`fp16`	True

Appendix C. Seen and unseen categories on Web NLG benchmark

Table 6

Experimental results distinguishing between seen and unseen categories on the Web NLG benchmark. 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. Scores of encoder-only models refer to results obtained with our hallucination-control heuristic (see Section 3.6.1).

		Web NLG			Seen Categories			Unseen Categories		
Arch.	**Family**	**Size**	**Pre-training**	**Task Learning**	**R-1**	**R-2**	**R-L**	**R-1**	**R-2**	**R-L**
Encoder-decoder	T5	60.5M		Fine-tuning	0.76	0.60	0.62	0.65	0.45	0.55
		223M		Fine-tuning	0.86	0.75	0.67	0.76	0.59	0.62
		738M		Fine-tuning	0.91	0.82	0.70	0.80	0.63	0.66
	BART	406M		Fine-tuning	0.79	0.63	0.61	0.70	0.53	0.59
		406M	REBEL	Fine-tuning	0.91	0.82	0.69	0.76	0.61	0.63
Decoder-only	Mistral-v0.1	7B		iCL + Heuristic	0.77	0.60	0.62	0.78	0.61	0.66
		7B	Conversation	iCL + Heuristic	0.76	0.56	0.60	0.76	0.55	0.62
		7B	OpenOrca	iCL + Heuristic	0.80	0.61	0.63	0.79	0.62	0.66
	Llama-2	7k	Instruction	iCL + Heuristic	0.76	0.55	0.59	0.74	0.52	0.60
		13B	Instruction	iCL + Heuristic	0.77	0.57	0.61	0.77	0.57	0.63
		7B	Meditron	iCL + Heuristic	0.72	0.53	0.58	0.71	0.51	0.59

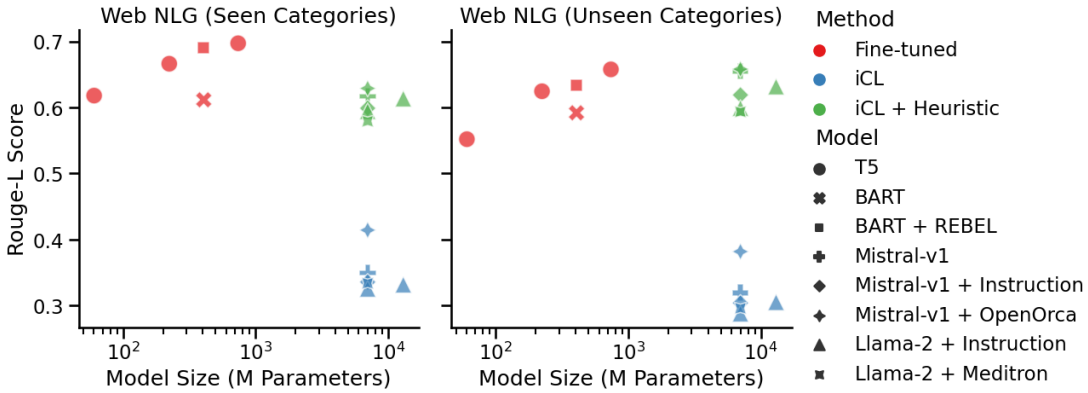

Fig. 12. Visualisation of Rouge-L scores in Table 6, in addition to the results of iCL without output being truncated by our hallucination-control heuristic. We observe, unsurprisingly, that performance in seen categories is slightly superior to unseen categories, resembling within-distribution generalisation as opposed to out-of-distribution generalisation. However, results display similar patterns, likely due to the atom and compound divergence between datasets discussed in Section 3.2.

Appendix D. Effect of number of in-context examples

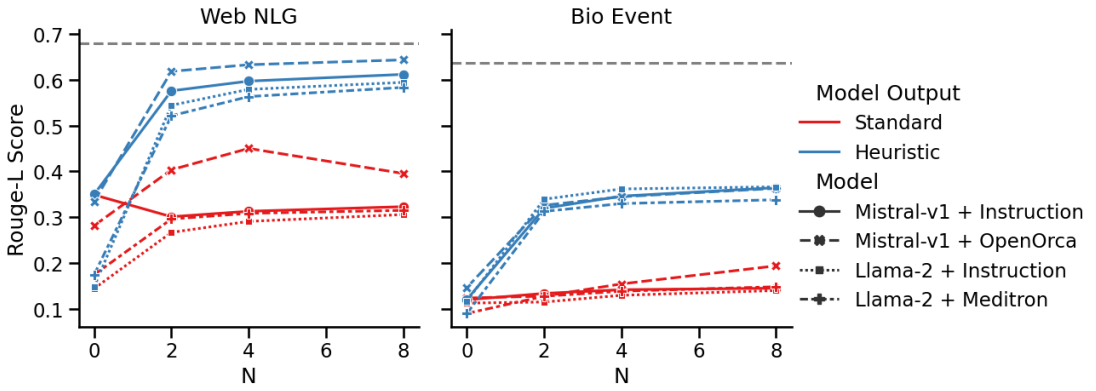

Fig. 13. Effect of in-context examples N on Rouge-L scores in decoder-only models, when evaluated with and without hallucination-control heuristic. The set of N is restricted compared to Figure 4, due to the smaller context window size of the Llama-2 family, but the patterns are similar.

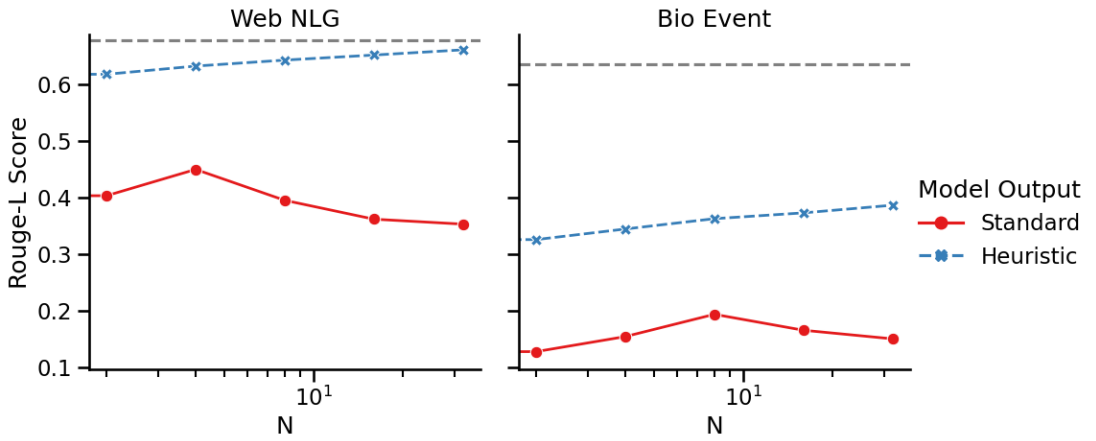

Fig. 14. Identical to Figure 4 but with the number of in-context examples N on a log-scale. We observe a log-linear relation between the performance behaviour and the number of in-context examples when our hallucination-control heuristic is considered.

Appendix E. Hallucination-control heuristic

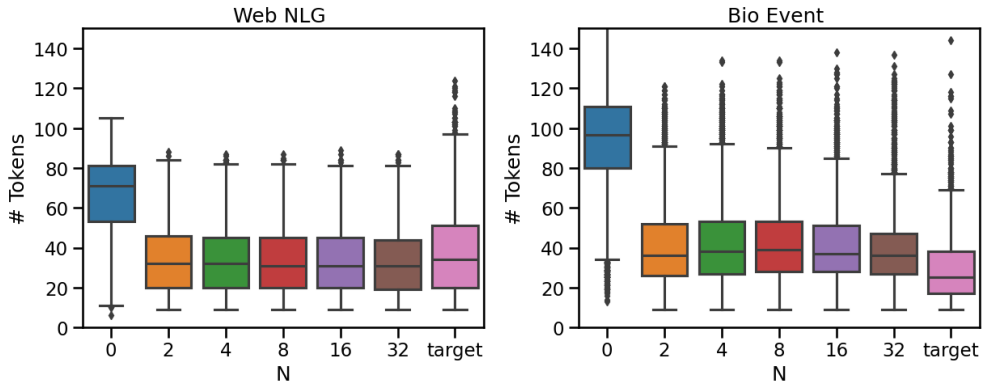

Fig. 15. Effect of the hallucination-control heuristic on the number of tokens outputted by the Mistral-v0.1 + OpenOrca model. For all N, except $N = 0$, the boxplots closely resemble the target distribution. In the case of $N = 0$, hardly any model output is truncated since the model does not follow the desired linearised graph structure.

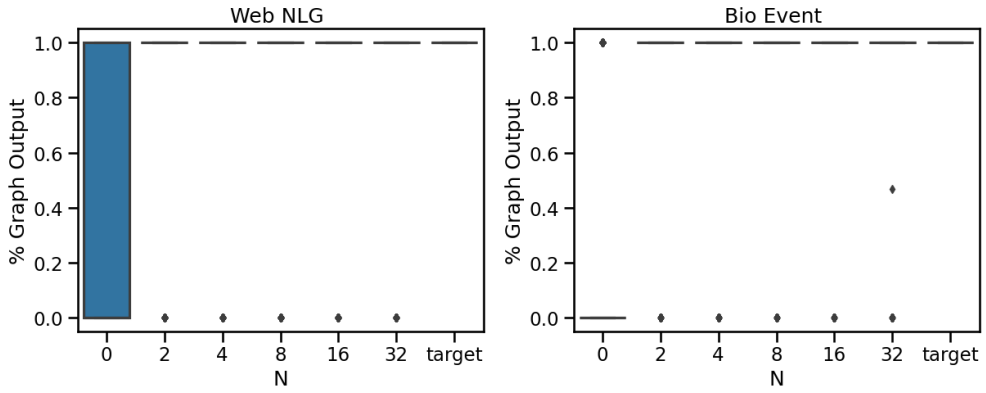

Fig. 16. Effect of the hallucination-control heuristic on the percentage of graph output, measured as the number of tokens inbetween "`[ (`" and "`) ]`" tokens, outputted by the Mistral-v0.1 + OpenOrca model. For all N, except $N = 0$, the boxplots closely resemble the target distribution, as all model output is a single graph. For $N = 0$, model output is mostly plain text.