Construction of Paired Knowledge Graph - Text Datasets Informed by Cyclic Evaluation

Ali Mousavi[*†], **Xin Zhan**[*†], **He Bai**[*†], **Peng Shi**[**‡], **Theo Rekatsinas**[†], **Benjamin Han**[†],
Yunyao Li[**§], **Jeff Pound**[†], **Josh Susskind**[†], **Natalie Schluter**[†], **Ihab F. Ilyas**[†], **Navdeep Jaitly**[†]
[†]Apple Inc, [‡] University of Waterloo, [§] Adobe Inc
{amousavi, xzhan, hbai22, thodrek, benjamin.b.han, pound, jsusskind, natschluter, iilyas, njaitly}@apple.com
yunyaol@adobe.com, peng.shi@uwaterloo.ca

Abstract

Datasets that pair Knowledge Graphs (KG) and text together (KG-T) can be used to train forward and reverse neural models that generate text from KG and vice versa. However models trained on datasets where KG and text pairs are not equivalent can suffer from more hallucination and poorer recall. In this paper, we verify this empirically by generating datasets with different levels of noise and find that noisier datasets do indeed lead to more hallucination. We argue that the ability of forward and reverse models trained on a dataset to cyclically regenerate source KG or text is a proxy for the equivalence between the KG and the text in the dataset. Using cyclic evaluation we find that manually created WebNLG is much better than automatically created TeKGen and T-REx. Informed by these observations, we construct a new, improved dataset called **LAGRANGE** using heuristics meant to improve equivalence between KG and text and show the impact of each of the heuristics on cyclic evaluation. We also construct two synthetic datasets using large language models (LLMs), and observe that these are conducive to models that perform significantly well on cyclic generation of text, but less so on cyclic generation of KGs, probably because of a lack of a consistent underlying ontology.

1. Introduction

The Natural Language Processing community has recently released several datasets with paired knowledge graphs (KG) and associated text (which we will refer to as KG-T) such as WebNLG (Gardent et al., 2017), TeKGen (Agarwal et al., 2021), KGPT (Chen et al., 2020) and T-REx (Elsahar et al., 2018). Such datasets can be used to train sequence-to-sequence models that can generate text from KGs (forward model) or vice versa (reverse model). However, prior studies assert that sequence-to-sequence models learn to hallucinate when the conditioning data has poor correlation with the sequence being produced, which can be the case when training data is noisy (Ji et al., 2023). In KG-text domain, hallucination can be quite problematic when the goal is to generate factually correct statements from KGs, in scenarios such as Question Answering.

When a KG-T evaluation dataset is available, it is easy to assess hallucination and recall of models trained on the data. For forward models, BLEU score between the text generated from the KG and the ground truth can be seen as a proxy for hallucination, while ROUGE score can be seen as a proxy for recall. For reverse models, comparing KG generated from the text, with the ground truth reveals how many KG facts are hallucinated, and how many are recalled. In Table 1 we show that as more and more noise is added to the KG part of WebNLG, which is manually created, the quality of text generated by forward models trained on it deteriorates, and so does the quality of the KG generated by the reverse models. Thus, a KG-T dataset which can be used to train reliable forward and reverse models needs to have as less noise as possible in the triples, and further, the information content between the text and the KG needs to be similar, as is the case with WebNLG.

However, most automatically generated datasets such as KGPT, TeKGen, T-REx have relatively sparse coverage of text with KG, since they are derived from existing KG datasets like Wikidata, whose coverage is relatively sparse. In these cases not only do models trained on these datasets hallucinate more, it is also hard to assess their accuracy on held out validation sets[1] because the validation sets themselves are highly noisy. Therefore, deciding on the best dataset for training these models can be challenging due to various factors, such as the peculiarities of the data KG ontology, the types of sentences found in different datasets, and so on. This makes it difficult to compare the results effectively.

In this paper, we claim that cyclic generation is a meaningful way of assessing the hallucination and recall of neural models trained on KG-T datasets, when a manually labelled set that has a comprehensive coverage of the text with KG is unavailable. In cyclic generation we start from one side (text or KG) and generate its counterpart (KG or text

[*]These authors contributed equally to this work.
[**]Work done while at Apple Inc.

[1]Which are also automatically created.

LREC-COLING 2024, pages 3782–3803
20-25 May, 2024. © 2024 ELRA Language Resource Association: CC BY-NC 4.0

Dataset	Graph-to-Text		Text-to-Graph	
	BLEU-4	ROUGE-4	Precision	Recall
WebNLG	44.59	31.30	90.00	89.40
+10% noise	44.46	31.44	89.79	88.76
+20% noise	43.97	30.96	89.43	88.29
+30% noise	43.54	30.54	88.16	87.28
+40% noise	42.56	29.92	87.05	86.83
+50% noise	41.56	28.73	83.65	85.51

Table 1: Models trained with noisy data.

respectively) using the appropriate (forward or reverse) sequence-to-sequence model trained on a KG-T dataset. The source is then regenerated using the model that works in the opposite direction. When we start from the graph, we call the cyclic reconstruction GTG; and when we start from the text, we call it TGT. GTG measures the ability to reproduce the KG with its specific ontological requirements while TGT measures the ability to reproduce data in more free formed text. Again, BLEU score between the original text and the reconstruction in TGT can be seen as a measure of hallucination, while the ROUGE score can be seen as a measure of recall. In GTG, triples can be matched more reliably than text to measure precision and recall of the facts, since the triples follow an ontology and comparison is less ambiguous than comparing free form text.

In these settings, cyclic evaluation is a better way of assessing which dataset to train the neural models, compared to assessing forward and reverse models separately, unidirectionally. This is because the evaluation does not rely on knowing ground truth matches, which are unavailable because the datasets are automatically constructed by alignment. Instead, it can rely on the sentences from the datasets, or the KG alone, separately. This is reminiscent of back-translation as being a way of assessing the quality of machine translation – since the assessment is performed on the known ground truth itself.

We use this method to compare several KG-T datasets and show that manually created WebNLG is much better than TeKGen and T-REx which are constructed automatically by aligning Wikipedia sentences with Wikidata. We use the lessons learnt to construct a new, **la**rge-scale **gra**ph-text alig**n**ed dataset for graph-text cross-modal **ge**neration (**LAGRANGE**) [2] using heuristics meant to improve alignment and coverage and show how each of the heuristics improves cyclic generation. We also construct two synthetic datasets using large language models (LLMs), and observe that these produce models that do very well on cyclic generation of text, but less so on cyclic generation of KGs. We hypothesize that this is probably because they lack a consistent ontology from one example to the next, which makes it difficult for neural models to reconstruct the exact KG through cyclic generation. This is meanwhile, not a problem for generating the text cyclically, since the neural network models are able to learn to deal with the variability in ontology, when reconstructing text from the "KG" of the dataset generated by the LLMs.

Finally, we also use GPT4 to compare the quality of our new dataset LAGRANGE and models trained based on it with other datasets and models trained based on them. In particular, we use GPT4 to measure the amount of hallucination or information missing in different datasets and models and find that the results agree with our cyclic evaluations.

2. Related Work

Prior surveys have reported on the impact of noisy data on hallucination (Ji et al., 2023) in sequence to sequence models, calling it "source-reference divergence". Other works have tried to reduce hallucination, for example by penalizing outputs that are hallucinations (Zhou et al., 2021). The use of cyclic generation is not new – it has been used as a way of improving generative models in KG-text settings (Wang et al., 2023; Guo et al., 2020). In contrast, we claim that a part of the reason why cyclic generation is poor, is because poor equivalence between KG and text in the dataset teaches the model to hallucinate missing facts. We thus propose to evaluate the quality of aligned graph-text datasets by measuring the cyclic generation abilities of models trained on them.

We briefly describe how our approach to create LAGRANGE is different from how other KG-T datasets were created. WebNLG (Gardent et al., 2017)) is a small scale manually created dataset and is thus of high quality but has a limited ontology. KGPT(Chen et al., 2020), GenWiki (Jin et al., 2020) and TeKGen(Agarwal et al., 2021) align Wikidata triples to the text in Wikipedia articles, by having different strategies for matching subjects or objects and hyperlinks in the text to Wikidata triples, but these methods do not check for semantic relevance of the KG to the sentence. In contrast, T-REx (Elsahar et al., 2018) utilizes predicate linker and coreference resolution to match KG triples to text, but may miss matches when the predicate is semantically entailed but not explicitly mentioned. See the Appendix A.3 for a more detailed explanation.

An important line of work focuses on new and improved models for Graph-Text conversion (Ribeiro et al., 2020). Our focus in this paper is not on building novel models but instead on reducing hal-

[2]Here are the links to training https://docs-assets.developer.apple.com/ml-research/datasets/lagrange/lagrange_train.json and test https://docs-assets.developer.apple.com/ml-research/datasets/lagrange/lagrange_test.json sets.

lucination and missing information by constructing a higher quality graph-text paired dataset on Wikidata-Wikipedia.

While the focus of this work is on creating a KG-T dataset based on triple-alignment, some related works have focused on relation extraction. In particular, REBEL (Cabot and Navigli, 2021) is a relation extraction dataset whose text might be redundant, while in this work we create a dataset for both G2T and T2G generation whose text and graph are highly aligned. REBEL (Cabot and Navigli, 2021) links the entities present in Wikipedia abstracts as hyperlinks while our dataset goes beyond hyperlinks and consider all entities in Wikipedia abstracts that have also appeared in Wikidata knowledge graph. This results in a considerable increase in the size of our dataset compared to REBEL.

3. Cyclic Evaluation of KG-T Datasets

A KG-T dataset is defined as a set of N paired (graph, text) tuples, $\{(\mathcal{G}_i, \mathcal{T}_i)\}_{i=1\cdots N}$ where each graph $\mathcal{G}_i$ is matched to a natural language sentence (or paragraph) $\mathcal{T}_i$. Here each graph, $\mathcal{G}_i$, is a set of K_i tuples $\{(s_j, p_j, o_j)\}_{j=1\cdots K_i}$ where each tuple describes a relationship (predicate) p_j between a subject s_j and an object o_j. [3]

We train the parameters θ of a model $G2T(\cdot;\theta)$ to predict the text $\mathcal{T}$ associated with the graph, $\mathcal{G}$, by minimizing a loss function $l(\cdot,\cdot)$,

$$\min_{\theta} \sum_{i=1}^{N} l(G2T(\mathcal{G}_i;\theta), \mathcal{T}_i) \quad (1)$$

Similarly, we also train a model $T2G(\cdot;\phi)$ to predict the graph $\mathcal{G}$ associated with the text $\mathcal{T}$ by minimizing an appropriate loss function. The models resemble the characteristics of a KG-T dataset and reflect the degree of hallucination and recall during generation. In other words, if the quality of a KG-Text dataset is lower, where the triple-text alignment contains precision and recall problems, it is possible that a trained model will learn such mistakes and reflect them in higher degree of hallucination and recall problems.

For cyclic evaluations (see Figure 1), we compute a GTG score, $s(\mathcal{G}', \mathcal{G})$, which compares a cyclically generated set of triples $\mathcal{G}' = T2G(G2T(\mathcal{G};\theta);\phi)$ against the original set of triples, $\mathcal{G}$. Similarly we compute a TGT evaluation by computing a score $s(\mathcal{T}', \mathcal{T})$ which compares a cyclically generated sentence $\mathcal{T}' = G2T(T2G(\mathcal{T};\phi);\theta)$ against the original $\mathcal{T}$.

[3] Note that we sometimes refer to the whole collection of triples in a dataset such as Wikidata as the KG, and the subset that is matched to a particular sentence in a KG-T dataset also as the KG. We use these interchangeably, but it should be obvious from context.

We can use different models, loss functions and scores for the assessment. In this paper, we trained transformer based T5 models, with the cross-entropy loss which is a sequence to sequence loss function. The quality of the results was assessed using various metric scores, including BLEU, ROUGE, and others (see section 5 for more details).

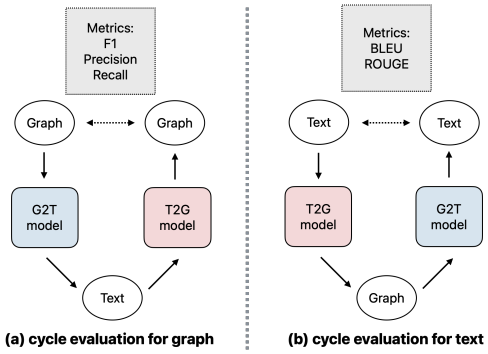

Figure 1: Two cycle evaluation processes.

4. Datasets

In this section, we describe the methodology used to create LAGRANGE and the synthetic datasets.

4.1. LAGRANGE

LAGRANGE consists of pairs of aligned KG triples from Wikidata and sentences from Wikipedia. We created an initial alignment between Wikidata KG triples, and Wikipedia using string matching techniques, and subsequently filtered out low quality matches using a semantic entailment model. Finally we augmented the KG triples by generating from a T2G model.

4.1.1. Generating an Initial Alignment

Albert Einstein was born in Ulm, in the Kingdom of Württemberg in the German Empire, on 14 March 1879 into a family of secular Ashkenazi Jews.

(Albert Einstein, Country of Citizenship, German Empire)
(Albert Einstein, Date of Birth, March 14th 1879)
(Albert Einstein, place of birth, Ulm)
(Albert Einstein, Ethnic Group, Jews)
(Albert Einstein, Given Name, Albert)
(Albert Einstein, Family Name, Einstein)
(Ulm, Country, Kingdom of Württemberg)
(Jews, Has Part, Ashkenazi Jews)

Figure 2: An example sentence from the Albert Einstein's Wikipedia article and the matched triples. Green colored words refer to the first-hop neighbors of Albert Einstein node on Wikidata and red colored words refer to its second-hop neighbors.

At a broad level, Wikidata can be described as a collection of KG triples (s, p, o), representing a relationship (referred to as a predicate), p, between a subject entity, s, and an object entity o. An initial pairing between Wikipedia sentences and Wikidata KG triples is easily achieved by matching the subject of the Wikipedia page containing the sentence to Wikidata triples about the same subject[4]. We additionally make sure that the subject or its aliases is explicitly referenced in the sentence. These initial matches are then filtered to remove KG triples where the object entity, or its alias is not matched to the sentence. The remaining triples are regarded as first-hop matches. Note that the actual dataset construction deals with corner cases of compound predicates in Wikidata, handling of dates and aliases, among other factors. A more detailed construction description is included in section A.1 as some of the details are not essential to understanding the main theme of the paper.

4.1.2. Incorporating Second-Hop Neighbors

A significant number of sentences contain additional information that does not relate to the subject entity, but to other entities in the sentence. In order to ensure a good coverage of the information present in the sentence, we also matched second-hop KG triples – which are triples whose subject is an entity that was an object in one of the triples in the first-hop alignment. As before, we ensure that the object entities of these second-hop triples are found in the sentence. See Figure 2 for an example of aligned text and its corresponding KG.

4.1.3. Improving Predicate Matching

The alignments generated in the previous section do not perform any verification that the text encapsulates the predicates of the triples matched to it which can lead to false matches where the triples contain information that is not present in the sentence. To fix this, we use an entailment model to remove aligned KG triples that were not entailed by the text (See Figure 3). We take a RoBERTa (Liu et al., 2019) model[5] fine-tuned on natural language inference (NLI) datasets, including SNLI (Bowman et al., 2015), ANLI (Nie et al., 2020) and MNLI (Williams et al., 2018) and feed in a sentence and a triple as input pairs. The entailment model produces an entailment score which predicts whether or not the sentence entails the facts described by the triple. KG triples which receive poor entailment scores are removed.

[4]These are nicely linked together through a unique identifier that Wikidata calls $\mathrm{Q_{id}}$.

[5]https://huggingface.co/ynie/roberta-large-snli_mnli_fever_anli_R1_R2_R3-nli

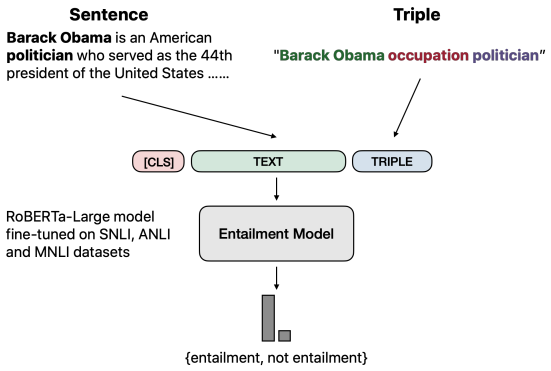

Figure 3: Alignment filter with entailment model.

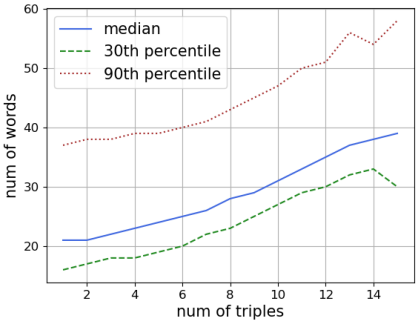

Figure 4: The relationship between the number of triples and the number of words in the sentence.

4.1.4. Ensuring Sufficient Coverage

The Wikipedia corpus contains a substantial number of lengthy sentences. Many of these sentences are only covered by a limited number of matching triples because Wikidata KG covers facts in Wikipedia quite sparsely. To mitigate this problem we remove examples whose sentence length appears to be longer in comparison to the number of available triples. To obtain a threshold we plotted the relationship between the length of sentences (measured in terms of the number of words) against the number of aligned triples (Figure 4). As observed, there exists a roughly linear relationship between the number of aligned triples and the number of words. We remove matches where the sentence length is greater than the 90^{th} percentile length among all examples with the same number of KG triples. However, for single-triple examples, we set a tighter length threshold at the 30^{th} percentile length, since the vast majority of examples in our dataset contain only a single triple.

4.1.5. Triple Augmentation

A lack of coverage of the sentence can also result from KG triples either being overlooked during our construction process or not being available on Wikidata. We mitigate this issue by generating additional triples from a T2G model trained on

Dataset	#Sent.	#Tri.	Avg.Tri.	Avg.Words
WebNLG	35K	104K	2.9	19.8
TeKGen	6.3M	10.9M	1.7	21.3
T-REx [6]	5.0M	13.1M	2.6	21.8
LAGRANGE	3.0M	12.3M	4.0	17.9
ChatGPT-GT	1.0M	4.1M	4.2	17.9
Guanaco-GT	2.7M	15.0M	5.6	17.7

Table 2: Training sets statistics. For a more detailed version, see Appendix A.4.

the data so far. The generated *new* triples are added as an augmentation to the original training set. This technique can be thought of as an analogue of back translation in neural machine translation (Edunov et al., 2018).This process can be potentially repeated in iterations until the generation results converge. In this paper, we only run it for one iteration as a demonstration.

4.2. Synthetic Datasets Using LLMs

As synthetic data generation using LLMs becomes widely adopted, it is interesting to understand the quality of the LLM-generated KG-text dataset using our evaluation framework. Due to the significant difficulty in compelling LLMs to generate KGs with canonical entity and predicate names from Wikidata, we relax our requirements and allow the triple elements to be open vocabulary. We prompt the LLMs to generate Wikidata style KG triples, with few-shot in-context examples as demonstration. We experiment with ChatGPT and Guanaco-33B (Dettmers et al., 2023) to generate graph from Wikipedia text with LLM instruction prompts. The generated datasets are referred to as Guanaco-GT and ChatGPT-GT. Given the throughput limitation of GhatGPT, we collected only 1 million examples for ChatGPT-GT. For Guanaco-33B we were able to generate 2.6 million examples which is on par with the size of LAGRANGE. More details of the LLM prompts and decoding configurations are provided in Appendix A.2.

5. Experiments and Discussions

In this section, we first introduce our experimental setup. Then, we show the results of three types of KG-T datasets: manually created, automatically constructed, and LLM generated. We then present an ablation study of our proposed techniques used to create the LAGRANGE dataset.

5.1. Setup

We treat both the G2T and T2G tasks as sequence-to-sequence modeling tasks in the experiments. More sophisticated approaches such as (Clive et al., 2022) can be applied under our evaluation framework, but we use a vanilla setup here for demonstration. To denote "subject", "predicate", "object", "qualifier", and "value" of a triple, we employ special tokens <S>, <P>, <O>, <Q>, and <T> respectively. The triples are connected by <sep> and serialized as a sequence. We fine-tune T5-large (Raffel et al., 2020) model on each dataset for our cycle-evaluation experiments. We use "`graph_to_text: `" as the T5 prefix for G2T and "`text_to_graph: `" for T2G. The models are trained with 8*A100 GPUs and the batch size is 48 for WebNLG and 192 for the others. We use AdamW (Loshchilov and Hutter, 2017) optimizer with the learning rate of 5e-05 and a linear decay learning rate scheduler. The total training steps are various across different dataset: 20K for WebNLG, 50K for LLM generated datasets, and 400K for the others. During decoding, we use the beam search size of 4.

5.2. Metrics

For GTG evaluation, we measure the quality of the reconstructed graph with the precision, recall, and F1 scores of triples. For each example, we count the number of triples in the reconstructed graph that also appears in the ground-truth graph, and then calculate the scores of each example. For TGT evaluation, we use the BLEU (Papineni et al., 2002) score and the ROUGE (Lin, 2004) score as metrics to evaluate the text regeneration.

5.3. Datasets

For evaluation, we take WebNLG v3 (Gardent et al., 2017) as an example of human annotated KG-T dataset. We use LAGRANGE, TeKGen (Agarwal et al., 2021) and T-REx (Elsahar et al., 2018) are examples of KG-T datasets created by automatic alignment. Finally, we use synthetic datasets generated by LLMs - ChatGPT-GT and Guanaco-GT by prompting. The statistics of these datasets are shown in Table 2. The test set size is 1.6K for WebNLG and 10K for the others. In Table 4, we show an example from each of the datasets. We have also provided examples of cyclic generation in Appendix A.6. Additionally, we provide the statistics of various versions of our LAGRANGE dataset in Appendix (Table 10), which will be elaborated upon in Section 5.6.

5.4. Effect of Noise in KG-T Data

Several factors, including dataset noise, graph complexity, and model performance, can all play a role in causing variations between the generated content and the reference content. However, our goal in this work is to marginalize the effect of dataset

[6] T-REx does not split between train and dev/test. We holdout 20% of the data for evaluation.

Dataset	Cycle TGT				Cycle GTG		
	BLEU-1	BLEU-4	ROUGE-1	ROUGE-4	F1	Precision	Recall
WebNLG	74.68	45.09	79.17	32.80	91.42	92.81	91.27
+10% noise	74.19	44.72	78.88	32.99	91.25	92.06	90.76
+20% noise	73.41	44.28	78.13	32.55	90.55	91.89	89.73
+30% noise	72.69	43.66	77.37	31.80	88.36	90.20	87.45
+40% noise	71.21	42.38	76.01	30.85	85.50	87.77	84.72
+50% noise	70.71	41.88	75.37	29.77	81.18	83.21	81.42

Table 3: Cyclic evaluation for WebNLG with different amount of noise.

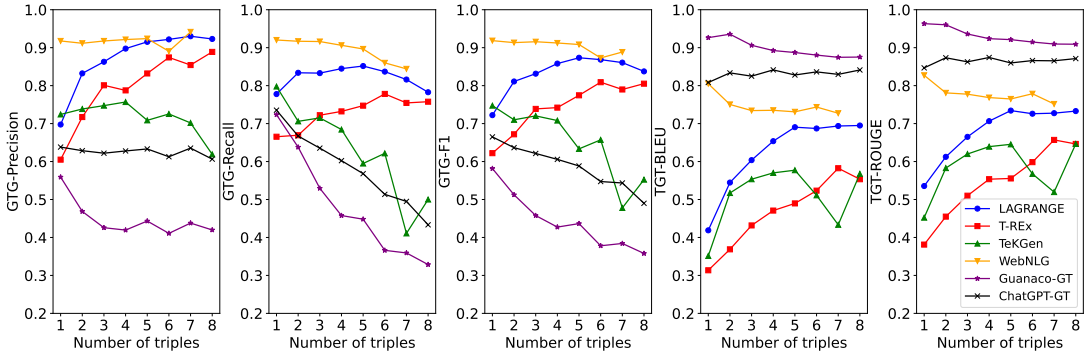

Figure 5: Cyclic evaluation broke down by the number of triples.

Dataset	Graph
LAGRANGE	(Kittie, has part or parts, Morgan Lander) (Morgan Lander, occupation, guitarist) (Morgan Lander, occupation, singer) (Kittie, instance of, musical group)
TeKGen	(Kittie, has part, Morgan Lander)
T-REx	(Kittie, has part, Morgan Lander) (Morgan Lander, member of, Kittie)
ChatGPT-GT	(Morgan Lander, occupation, lead vocalist) (Morgan Lander, occupation, guitarist) (Morgan Lander, band, Kittie) (Tanya Candler, occupation, bassist) (Kittie, member, Morgan Lander) (Kittie, member, Tanya Candler)
Guanaco-GT	(Morgan Lander, became, lead vocalist) (Morgan Lander, became, one of Kittie's guitarists) (Tanya Candler, completed, the band's lineup on bass) (Kittie, lineup, Morgan Lander lead vocalist one of Kittie's guitarists Tanya Candler bass)

Table 4: Examples of KG triples aligned by different datasets for sentence: "*Morgan Lander became the lead vocalist and one of Kittie's guitarists and Tanya Candler completed the band's lineup on bass.*". See Appendix A.5 for more examples.

noise. For that reason, we have kept other factors constant in the experiments. All automatically generated datasets are based on Wikidata graph and hence have the same graph complexity. Also, all models are fintuned from the same initial checkpoint (T5-large) to make sure they have the same performance before the finetuning.

We first test our assertion that cyclic evaluation results are reflective of noise in the datasets. To do this, we modify the WebNLG dataset by randomly inserting, deleting, or substituting triples in the examples. We can control the level of misalignment by controlling the probability with which triples are modified[7]. We observe that as additional noise is introduced, the forward and reverse models trained on these noisy datasets become less accurate in both precision and recall, which confirms the "source-reference divergence" hypothesis (see Table 1 for unidirectional evaluations and Table 3 for cyclic evaluations). One might assume that neural models can deal with noise in the input, but these results indicate that the quality of models does suffer with more noise. Comparing unidirectional results with cyclic results, we see that if there is any noise is the datasets, cyclic evaluation reveals a more intensified version of it compared to unidirectional evaluation. This indicates the possibility that cyclic evaluations might have better resolution than unidirectional evaluations, since they can assess the effect of errors made in the unidirectional generations, on the reconstruction of the original source. Finally, we note again that cyclic evaluation assesses the dataset quality without the ground-truth label (with only the text or the KG) and hence can be used to compare different datasets and ontologies.

5.5. Main Results

The cyclic evaluation results of different datasets are shown in Table 5.

Manually created dataset. We can first see that WebNLG is of much higher quality: it gets the best GTG cyclic evaluation results with 91.41 F1

[7]We assume that WebNLG is mostly free of noise since it was constructed manually.

Dataset	Cycle TGT				Cycle GTG		
	BLEU-1	BLEU-4	ROUGE-1	ROUGE-4	F1	Precision	Recall
WebNLG	74.68	45.09	79.17	32.80	**91.42**	**91.81**	**91.27**
TeKGen	44.10	28.11	51.99	23.09	73.88	74.19	75.92
T-REx	38.65	21.39	45.50	16.92	67.50	69.80	68.60
LAGRANGE	63.38	47.46	67.50	38.96	84.33	87.11	84.60
Guanaco-GT	**88.75**	**77.99**	**92.13**	**75.18**	41.48	42.52	43.56
ChatGPT-GT	83.13	68.78	86.55	63.03	58.30	62.23	57.58

Table 5: Cyclic evaluation results of manually created dataset, automatically constructed datasets, and LLM generated datasets.

Dataset	Cycle TGT				Cycle GTG		
	BLEU-1	BLEU-4	ROUGE-1	ROUGE-4	F1	Precision	Recall
V0	47.11	31.68	55.50	27.74	80.91	82.94	82.82
V1 (V0+semantic filter)	49.69	33.93	57.08	29.12	81.16	82.68	83.51
V2 (V1+second hop)	49.81	34.64	57.53	30.00	78.73	80.96	81.06
V3 (V2+length filter)	62.27	46.31	66.27	37.78	82.17	85.63	82.30
LAGRANGE (V3+augment)	**63.38**	**47.46**	**67.50**	**38.96**	**84.33**	**87.11**	**84.60**

Table 6: Ablation study.

Dataset	Cycle TGT with TeKGen Text				Cycle TGT with LAGRANGE Text			
	BLEU-1	BLEU-4	ROUGE-1	ROUGE-4	BLEU-1	BLEU-4	ROUGE-1	ROUGE-4
T-REx	35.74	18.78	42.78	13.75	47.88	27.33	50.86	17.29
TeKGen	44.10	28.11	51.99	23.09	58.25	39.36	62.94	30.93
LAGRANGE	**44.36**	**28.63**	**53.14**	**25.43**	**63.38**	**47.46**	**67.50**	**38.96**

Table 7: TGT results evaluated with TeKGen and LAGRANGE, respectively.

scores, and 74.68 BLEU-1 scores for TGT cyclic evaluation. The results confirm that WebNLG is a well-aligned dataset. However, it is important to note that WebNLG has only 35k data points with a limited ontology that only covers approximately 600 entities and 20 relationships.

Automatically constructed datasets. LAGRANGE gets the best results among the datasets created using automatic alignment methods, with 84.33 GTG F1 scores, 63.38 TGT BLEU-1, and 47.46 TGT BLEU-4. LAGRANGE demonstrates superior alignment between the text and graph compared to TeKGen and T-REx.

It is worth mentioning that LAGRANGE dataset contains a larger number of triples than the other datasets (Table 2). LAGRANGE achieves higher precision and recall in KG reconstruction (GTG). In addition, LAGRANGE's 4-gram TGT results are better than WebNLG because WebNLG aligns multiple sentences to a set of triples. The introduction of sentence order in WebNLG introduces additional errors for 4-gram evaluations.

LLM generated datasets. Guanaco-GT and ChatGPT-GT demonstrate significant superiority over the others in terms of TGT BLEU and ROUGE scores, but not in GTG evaluation. In other words, the text is exceptionally well-reconstructed in TGT cyclic evaluation. This can be attributed to the fact that LLMs have the ability to invent new predicates and entities, enabling them to describe relations and facts that cannot be represented by Wikidata triples. However, this freedom also allows LLMs to generate non-existing or redundant facts and create meaningless or incoherent triples, which significantly limits the usability of the datasets. The GTG cyclic evaluation reveals that the triples generated by LLMs are not as reproducible as those produced by LAGRANGE and other KG-grounded datasets. We observed that Guanaco-GT performs notably better than ChatGPT-GT in TGT evaluation. This is likely due to the Guanaco model's tendency to parse input sentences into multiple phrases, making sentence reconstruction easier. While the ChatGPT model also suffers from this issue, its severity is relatively lower when compared to Guanaco. This explains why ChatGPT outperforms Guanaco in GTG but not in TGT. An example is shown in Table 4.

Finally, we visualize the cyclic evaluation results by segmenting the evaluation dataset based on the number of triples. As illustrated in Figure 5, LAGRANGE consistently surpasses TeKGen and T-REx in performance. It is worth noting that as the number of triples increases, LAGRANGE experiences a slight decrease in recall, but its precision improves, with a more consistent F1 score, while TeKGen and LLM generated datasets decline in both the precision and recall. Also, it is important

to highlight that while the LLM-generated datasets yield better TGT evaluation results, their GTG evaluation results are the worst.

5.6. Ablation Study of LAGRANGE

We further conducted an ablation study on our proposed techniques for constructing the LAGRANGE dataset. The results are presented in Table 6. As observed, all the proposed techniques consistently resulted in improvements across almost all metrics for both TGT and GTG evaluations, affirming their effectiveness. However, there was one exception: the V2 performance for GTG. This can be attributed to an imbalance between triples based on first-hop and second-hop neighbors. Since there are more first-hop-based triples, we observed a slight decline in precision and recall for GTG.

Meanwhile, the most significant performance gain was achieved through the *length filtering* step. This can be intuitively explained by the fact that regardless of the techniques employed, it is impossible to generate sentence segments for which corresponding triples are lacking. Hence, the application of length filtering enhances the feasibility of sentence-graph generation.

5.7. Unified Comparison

Finally, we evaluate all models using the same test data for TGT evaluation. In particular, we use the TeKGen evaluation text and the LAGRANGE evaluation text, respectively.

The results are presented in Table 7. It is evident that the model trained with LAGRANGE achieve the highest BLEU and ROUGE scores in both cases. In particular, we observe that the model trained with LAGRANGE data and evaluated with TeKGen data outperform the model trained on TeKGen itself, demonstrating the effectiveness and adaptability of our evaluation methodology across different text styles.

In addition to BLEU and ROUGE metrics that primarily focus on n-gram matching, we measure the quality of generated sentences in terms of their equivalence to the original sentences. To achieve this goal, we ask GPT4 to rank generated sentences in terms of their equivalence to the original sentence. We use chain-of-thought style prompting (Wei et al., 2022) (see Appendix A.7) and define equivalence as having the least amount of missing information or hallucination. We randomly sample 1000 original sentences, run them through models trained on different datasets and give the output of each model to GPT4 for final ranking. Figure 6 and 7 shows the rankings for models tested on LAGRANGE and TekGen test data, respectively. In both figures, the model trained with LAGRANGE is showing the best performance.

Finally, we use GPT4 to compare the original set of triples in each dataset with the generated sentence from its corresponding trained GT model. In particular, we use chain-of-thought style prompting (Wei et al., 2022) (see Appendix A.7) and ask GPT4 to generat a score between -10 to +10 where -10 denotes extreme miss of information, 0 means complete equivalence, and +10 means extreme hallucination. If a generated sentence suffers from both hallucination and missing information, we ask GPT4 to consider the more severe problem and output a score based on that. We randomly sample 1000 pairs of `(triples, generated sentence)` for each dataset. Figure 8 shows the distribution of non-negative scores (i.e., equivalence or hallucination) and Figure 9 shows the distribution of negative scores (i.e., missing information). Based on these two figures, the GT model trained on LAGRANGE has the best performance in terms of equivalence between the original set of triples and the generated sentences.

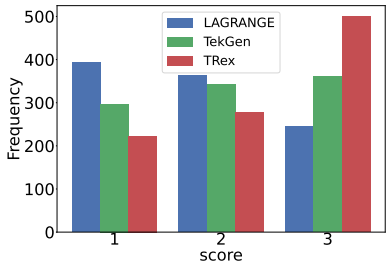

Figure 6: GPT4 ranking of TGT results evaluation with the LAGRANGE test set.

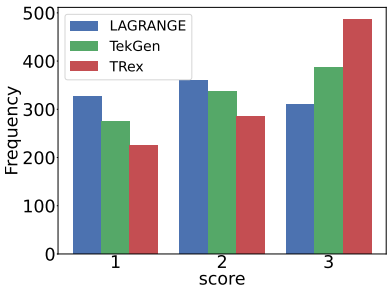

Figure 7: GPT4 ranking of TGT results evaluation with the TekGen test set.

6. Conclusion

In this paper, we have addressed the alignment problem between KG and text datasets. Our study focused on evaluating the alignment between KG triples and sentences, which lead us to propose a novel evaluation methodology that leverages the

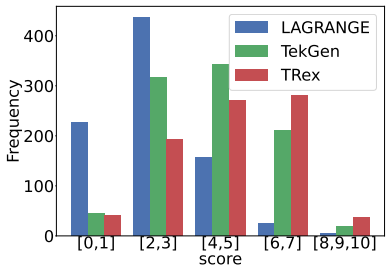

Figure 8: Non-negative GPT4 assigned scores to GT models outputs. The model trained with LAGRANGE has the least amount of hallucination.

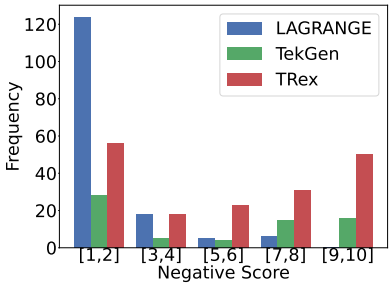

Figure 9: Negative GPT4 assigned scores to GT models outputs. The model trained with LAGRANGE has the least amount of missing information.

cyclic generation of the KG and of the sentences. Using this methodology, we were able to assess the quality of alignment in existing KG-text datasets and introduced a series of techniques to enhance dataset alignment.

We also introduced the LAGRANGE dataset, showcasing significant improvements in alignment compared to existing automatically collected KG-text datasets, using cyclic evaluation.

Finally, we created synthetic datasets using LLMs and evaluated the alignment of these LLMs-generated datasets, highlighting the advantages and disadvantages of such an approach.

To foster further research in this area, we make the LAGRANGE dataset publicly available. We believe that this resources will serve as valuable assets for the research community to explore and advance the field of KG-text integration.

Ethics Statement

In this paper, we propose a method to assess the alignment of KG-text datasets and a series of novel techniques to improve the alignment. There are more traditional alignment techniques that are not considered in this work (such as the ones used in (Elsahar et al., 2018)) since those are not the focus of this paper. LAGRANGE is based on WikiData and Wikipedia, which might not generalize well to other text domains or ontologies. Given the uneven distribution of demographics in Wikipedia corpus, LAGRANGE might inherit the bias and fairness issues (such as gender, race, occupation, etc.) from Wikipedia. Furthermore, although we have improved the alignment quality significantly, misaligned triples and phrases as well as the hallucination issue still exist to some extent. The last but not the least, triple alignment could be further improved by regenerating the sentence in each example. We do not consider that approach in this work since we prefer to keep the sentences natural.

Oshin Agarwal, Heming Ge, Siamak Shakeri, and Rami Al-Rfou. 2021. Knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3554–3565, Online. Association for Computational Linguistics.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Pere-Lluís Huguet Cabot and Roberto Navigli. 2021. Rebel: Relation extraction by end-to-end language generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2370–2381.

Wenhu Chen, Yu Su, Xifeng Yan, and William Yang Wang. 2020. KGPT: Knowledge-grounded pre-training for data-to-text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8635–8648, Online. Association for Computational Linguistics.

Jordan Clive, Kris Cao, and Marek Rei. 2022. Control prefixes for parameter-efficient text generation. In *Proceedings of the 2nd Workshop*

on Natural Language Generation, Evaluation, and Metrics (GEM), pages 363–382, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. *arXiv preprint arXiv:1808.09381*.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. 2018. T-rex: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating training corpora for NLG micro-planners. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 179–188, Vancouver, Canada. Association for Computational Linguistics.

Qipeng Guo, Zhijing Jin, Xipeng Qiu, Weinan Zhang, David Wipf, and Zheng Zhang. 2020. Cyclegt: Unsupervised graph-to-text and text-to-graph generation via cycle training. *arXiv preprint arXiv:2006.04702*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12).

Zhijing Jin, Qipeng Guo, Xipeng Qiu, and Zheng Zhang. 2020. 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*, pages 2398–2409.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial nli: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Leonardo FR Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. 2020. Investigating pretrained language models for graph-to-text generation. *arXiv preprint arXiv:2007.08426*.

Luyu Wang, Yujia Li, Ozlem Aslan, and Oriol Vinyals. 2021. WikiGraphs: A Wikipedia text - knowledge graph paired dataset. In *Proceedings of the Fifteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-15)*, pages 67–82, Mexico City, Mexico. Association for Computational Linguistics.

Zhuoer Wang, Marcus Collins, Nikhita Vedula, Simone Filice, Shervin Malmasi, and Oleg Rokhlenko. 2023. Faithful low-resource data-to-text generation through cycle training. *arXiv preprint arXiv:2305.14793*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Chunting Zhou, Graham Neubig, Jiatao Gu, Mona Diab, Francisco Guzmán, Luke Zettlemoyer, and Marjan Ghazvininejad. 2021. Detecting hallucinated content in conditional neural sequence generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1393–1404.

A. Appendix

A.1. Generating an Initial Alignment between Wikipedia sentences and Wikidata triples

In this section, we formally describe the methodology used to align Wikipedia entries to Wikidata KG triples, to create the initial version of LAGRANGE.

Wikipedia and Wikidata pages are associated with each other through a unique Q_{id}. For each Q_{id} $\mathtt{q}$, we consider the set of all sentences $\mathcal{T}_{\mathtt{q}}$ from its Wikipedia page. We also consider the set of all triples from the corresponding Wikidata page as $\mathcal{G}_{\mathtt{q}} = \{(s, p, o, q, t)\}$ where s denotes the subject or $\mathtt{q}$ title, p denotes the predicate, o denotes the object, q denotes the qualifier of the predicate p which is null for simple predicates, and t denotes the value of the qualifier which is also null if the qualifier q is null. In other words, the predicate p could either be simple as in `(Albert Einstein, Occupation, Scientist)` or compound as in `(Albert Einstein, Award Received, Nobel Prize in Physics, Point in Time, 1921)`. If the predicate of a triple has a qualifier, we consider it as a compound predicate and present its qualifier and corresponding qualifier's value as well. For the sake of simplicity in our discussion, we will explain the construction process based on simple predicates. However, it's important to note that the same principles apply to compound predicates as well.

For each sentence in $\mathcal{T}_{\mathtt{q}}$, our goal is to match as many triples as possible from $\mathcal{G}_{\mathtt{q}}$ and neighboring Wikidata graphs to it. Before describing the matching process, we would like to provide more details on how $\mathcal{G}_{\mathtt{q}}$ is built.

For each Q_{id} $\mathtt{q}$, the subject s in each triple $(s, p, o) \in \mathcal{G}_{\mathtt{q}}$ is always the title of $\mathtt{q}$ (e.g. *Albert Einstein* for Q_{937}). The object o on the other hand, could be

- the title or an alias of another Q_{id}. As an example, *Elsa Einstein* that is the title of Q_{68761} as in `(Albert Einstein, Spouse, Elsa Einstein)`.
- strings that are not associated to any Q_{id}. As an example, *2* as in `(Albert Einstein, Erdős number, 2)`.
- literal values such as dates or quantities and their possible aliases. As an example, we have `(Albert Einstein, date of birth, +1879-03-14T00:00:00Z)` on Wikidata. However, on a Wikipedia sentence, this date could be expressed as `March 14, 1879` or `March 14th, 1879`. Hence, we consider different aliases of a literal in order to match more triples.

For each sentence $\mathtt{s}$ in $\mathcal{T}_{\mathtt{q}}$, we define $\mathcal{M}'_{\mathtt{s},\mathtt{q}}$ as the set of all triples $(s, p, o) \in \mathcal{G}_{\mathtt{q}}$ where o has appeared in the sentence $\mathtt{s}$. In other words, $\mathcal{M}'_{\mathtt{s},\mathtt{q}}$ denotes the set of triples matched to $\mathtt{s}$ in which objects are first-hop neighbors of $\mathtt{q}$ in Wikidata graph. As an example, if the sentence $\mathtt{s}$ is [*Albert Einstein was born in Ulm, in the Kingdom of Württemberg in the German Empire, on 14 March 1879 into a family of secular Ashkenazi Jews.*], then `(Albert Einstein, Place of Birth, Ulm)` $\in \mathcal{M}'_{\mathtt{s},\mathrm{Q}_{937}}$ since the word `Ulm` has appeared in the sentence $\mathtt{s}$ and the entity `Ulm` (i.e., Q_{3012}) is a first-hop neighbor of `Albert Einstein` (i.e., Q_{937}) on Wikidata. It is noteworthy to mention that at this stage we do not put any constraint on the predicate of matched triples. We will later explain how our post-processing helps us to match predicates as well.

Having constructed the $\mathcal{M}'_{\mathtt{s},\mathtt{q}}$, let $\mathcal{Q}$ denote the set of all Q_{id}s in Wikidata and define $Q_{\mathcal{M}'_{\mathtt{s},\mathtt{q}}} = \{o | (s, p, o) \in \mathcal{M}'_{\mathtt{s},\mathtt{q}} \wedge \exists q' \in \mathcal{Q} : o \equiv q'\}$. In other words, for each sentence $\mathtt{s}$ in $\mathcal{T}_{\mathtt{q}}$, $Q_{\mathcal{M}'_{\mathtt{s},\mathtt{q}}}$ denotes the set of all Q_{id}s that have appeared in $\mathtt{s}$ in the original or an alias format and are first-hop neighbors of $\mathtt{q}$ in Wikidata. In the aforementioned example where the matched triple `(Albert Einstein, Place of Birth, Ulm)` is in $\mathcal{M}'_{\mathtt{s},\mathrm{Q}_{937}}$, the Q_{id} corresponding to `Ulm` (i.e., Q_{3012}) is also in $Q_{\mathcal{M}'_{\mathtt{s},\mathrm{Q}_{937}}}$.

In addition to $\mathcal{M}'_{\mathtt{s},\mathtt{q}}$, we define $\mathcal{M}''_{\mathtt{s},\mathtt{q}}$ as the set of all triples (s', p', o') where $s' \in Q_{\mathcal{M}'_{\mathtt{s},\mathtt{q}}}$ (i.e., subject s' is a Q_{id} and a first-hop neighbor of $\mathtt{q}$) and o' has appeared in the sentence $\mathtt{s}$. As an example, consider the following sentence $\mathtt{s}$ from Albert Einstein's Wikipedia article: [*Albert Einstein was born in Ulm, in the Kingdom of Württemberg in the German Empire, on 14 March 1879 into a family of secular Ashkenazi Jews.*]. Based on Albert Einstein's Wikidata graph, we have `(Albert Einstein, Place of Birth, Ulm)` $\in \mathcal{M}'_{\mathtt{s},\mathrm{Q}_{937}}$ and hence, $\mathrm{Q}_{3012}(\mathrm{Ulm}) \in Q_{\mathcal{M}'_{\mathtt{s},\mathrm{Q}_{937}}}$. Since `Ulm` is the first-hop neighbor of `Albert Einstein` and `Kingdom of Württemberg` is the first-hop neighbor of `Ulm`, then `(Ulm, Country, Kingdom of Württemberg)` is in $\mathcal{M}''_{\mathtt{s},\mathrm{Q}_{937}}$. Figure 10 shows the example sentences from Wikipedia and their corresponding subgraphs from Wikidata. In addition, Figure 2 shows the matched triples to each sentence based on the Wikidata subgraphs in Figure 10. As seen in Figure 2, considering second-hop neighbors can significantly increase the number of matched triples and give us a richer dataset for training graph-to-text generative models.

Once we have both $\mathcal{M}'_{\mathtt{s},\mathtt{q}}$ and $\mathcal{M}''_{\mathtt{s},\mathtt{q}}$, then we build $\mathcal{M}_{\mathtt{s},\mathtt{q}} = \mathcal{M}'_{\mathtt{s},\mathtt{q}} \cup \mathcal{M}''_{\mathtt{s},\mathtt{q}}$ to denote the set of all matched triples from $\mathcal{G}_{\mathtt{q}}$ to $\mathtt{s}$. Our raw dataset consists of 37 million sentences and 104 million triples. It is

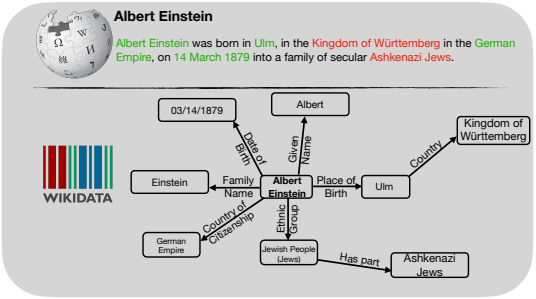

Figure 10: A sample sentence and its corresponding subgraph from Albert Einstein's Wikipedia article and Wikidata. Words that are annotated in green and red are first-hop and second-hop neighbors of Albert Einstein on Wikidata, respectively.

noteworthy to mention that 30 million triples in our initial dataset our based on second-hop neighbors of entities. Although our post-processing step filters out a number of these triples, we would like to emphasize on how going beyond first-hop neighbors can give us a dataset with higher coverage of information in the matched KG triples, unlike other datasets that we have mentioned them in Table 2.

A.2. Generating Synthetic Datasets from LLMs

We provide the LLMs prompt for synthetic datasets generation in Table 8. Since ChatGPT has a long context window, we provide few-shot examples in the prompt. The context window size of Guanaco is 2048, in order to keep the prompt concise, we show only one example in the prompt. We decode with greedy decoding so that the generation can be more stable. We do not claim these are the optimal prompts for the synthetic dataset generation task. There are rooms for tuning the prompt to improve the quality of the datasets.

LLM	Prompt
ChatGPT	Extract facts from a sentence in the form of tuples: 1. Each fact consists of a subject, a predicate, and an object. The fact might have a predicate_attribute and an attribute_value as well. 2. For predicates without any attribute, extract triples in form of (Subject, Predicate, Object) Sentence: John is an engineer living in Chicago. Triples: (John, occupation, engineer) (John, residence, Chicago) Sentence: There exists an actor called Simon Pegg. Triples: (Simon Pegg, occupation, actor) 3. For predicates with an attribute, extract one tuple per attribute in the form of (Subject, Predicate, Object, Predicate_attribute, Attribute_value) where Predicate_attribute is the attribute's name, and Attribute_value is the attribute's value. Sentence: John started working at Apple since 2008. Triples: (John, employer, Apple, start time, 2008) Sentence: Sara and Bob got divorced at 2012. Triples: (Sara, spouse, Bob, end time, 2012) (Sara, spouse, Bob, end cause, divorce) Sentence: {sentence} Triples:
Guanaco-33B	Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request. ### Instruction: Extract WikiData knowledge graph triples from the following sentence. For example, Sentence: Obama was elected over Republican nominee John McCain in the presidential election and was inaugurated on January 20, 2009. Triples: (Obama, elected over, John McCain) (Obama, inaugurated on, January 20, 2009) (Obama, election type, presidential election) (Obama, nominee, Republican nominee) (Obama, election date, January 20, 2009) ### Human: Sentence: {sentence} ### Assistant:

Table 8: The LLM prompts used for synthetic datasets generation.

A.3. Comparison between construction techniques of different datasets

Here we also contrast other approaches to create KG-T datasets with the method we used to create LAGRANGE in Table 9. WebNLG (Gardent et al., 2017)) is a small scale manually created dataset and is thus of high quality but has a limited ontology. TEKGEN(Agarwal et al., 2021) is a recently developed dataset that aligns Wikidata triples to the sentences in the first section of the corresponding Wikipedia articles, based on the presence of the triple object or its aliases in the sentence. However, does not assess whether predicates in the graph match appropriately with the sentence. KGPT(Chen et al., 2020), on the other hand, relies on unigram overlaps between sentences and graphs using Wikipedia hyperlinks, which can lead to missing matches for non-hyperlinked entities. GenWiki (Jin et al., 2020) is similar to KGPT, but smaller, and aligns triples and sentences based on entity set overlaps without predicate matching. In contrast, T-REx utilizes predicate linker and coreference resolution to match triples to sentences, but can miss matches when the predicate is semantically entailed in a sentence but not explicitly mentioned. WikiGraphs (Wang et al., 2021), unlike T-REx and other previously mentioned datasets, matches entire Wikipedia articles instead of individual sentences to Freebase KG (Bollacker et al., 2008), and it does not rely on predicate matching.

Table 9: Comparing different attributes of LAGRANGE with other datasets.

PROPERTY	LAGRANGE	TEKGEN	KGPT	GENWIKI	WIKIGRAPHS	T-REX	WEBNLG
HUMAN MADE	×	×	×	×	×	×	√
SECOND-HOP COVERAGE	√	×	×	×	×	×	×
NON-HYPERLINKED ANNOTATION	√	√	×	√	√	√	×
PREDICATE MATCHING	√	×	×	×	×	√	√
SEMANTIC ALIGNMENT	√	×	×	×	×	×	√

A.4. Datasets Statistics

Dataset	#Sent.	#Tri.	#Tri./Sent.	#Words
WebNLG	35K	104K	2.96	19.83
TeKGen	6.3M	10.9M	1.73	21.26
T-Rex	5.0M	13.1M	2.61	21.83
V0	5.2M	14.2M	2.70	20.38
V1(V0+semantic filter)	4.2M	10.0M	2.40	20.78
V2(V1+second hop triples)	4.3M	13.0M	3.00	20.77
V3(V2+length filtering)	3.0M	11.0M	3.59	17.90
LAGRANGE(V3+augment)	3.0M	12.3M	4.02	17.90
ChatGPT-GT	1.0M	4.1M	4.17	17.90
Guanaco-GT	2.7M	15.0M	5.59	17.72

Table 10: Statistics of the number of sentences, number of triples, and number of triples per sentence.

A.5. Dataset Examples

Text	Dataset	Graph
Uncommon Women and Others (1977), is the first play by noted 20th-century American playwright Wendy Wasserstein.	LAGRANGE	(Uncommon Women and Others, instance of, play) (Uncommon Women and Others, author, Wendy Wasserstein) (Wendy Wasserstein, occupation, playwright)
	TeKGen	(Uncommon Women and Others, author, Wendy Wasserstein)
	T-REx	(Wendy Wasserstein, occupation, playwright) (Wendy Wasserstein, country of citizenship, United States of America) (Uncommon Women and Others, author, Wendy Wasserstein)
	ChatGPT-GT	(Wendy Wasserstein, notable for, 20th-century American playwright) (Uncommon Women and Others, type, play) (Uncommon Women and Others, year, 1977) (Uncommon Women and Others, author, Wendy Wasserstein) (Uncommon Women and Others, first, true)
	Guanaco-GT	(Uncommon Women and Others, written by, Wendy Wasserstein) (Uncommon Women and Others, first play, 1977) (Uncommon Women and Others, 20th-century, American playwright Wendy Wasserstein) (Uncommon Women and Others, noted, 20th-century American playwright Wendy Wasserstein) (Uncommon Women and Others, play, Uncommon Women and Others)
Stewart's Restaurants are classic 1950s style fast-food restaurants located throughout the United States.	LAGRANGE	(Stewart's Restaurants, instance of, restaurant) (Stewart's Restaurants, country, United States of America)
	TeKGen	(Stewart 's Restaurants, country, United States)
	T-REx	(Stewart's Restaurants, country, United States of America)
	ChatGPT-GT	(Stewart's Restaurants, type, fast-food restaurant) (Stewart's Restaurants, style, 1950s) (Stewart's Restaurants, location, United States)
	Guanaco-GT	(Stewart's Restaurants, style, 1950s) (Stewart's Restaurants, location, United States) (Stewart's Restaurants, type, fast-food) (Stewart's Restaurants, date, classic) (Stewart's Restaurants, location, throughout)
The 2009 CAF Champions League is the 45th edition of Africa's premier club football tournament organized by the Confederation of African Football (CAF), and the 13th edition under the current CAF Champions League format.	LAGRANGE	(Confederation of African Football, operating area, Africa) (2009 CAF Champions League, organizer, Confederation of African Football) (2009 CAF Champions League, sport, association football) (Confederation of African Football, short name, CAF) (2009 CAF Champions League, point in time, 2009) (2009 CAF Champions League, sports season of league or competition, CAF Champions League)
	TeKGen	(2009 CAF Champions League, point in time, 00 2009)
	T-REx	(2009 CAF Champions League, instance of, CAF Champions League) (CAF Champions League, sport, association football) (2009 CAF Champions League, sport, association football) (Confederation of African Football, sport, association football)
	ChatGPT-GT	(2009 CAF Champions League, edition number, 45th) (2009 CAF Champions League, tournament name, Africa's premier club football tournament) (2009 CAF Champions League, organizer, Confederation of African Football) (2009 CAF Champions League, tournament format, CAF Champions League) (2009 CAF Champions League, current edition number, 13th)
	Guanaco-GT	(2009 CAF Champions League, edition, 45th) (2009 CAF Champions League, format, CAF Champions League) (2009 CAF Champions League, format, 13th edition) (2009 CAF Champions League, organization, Confederation of African Football) (2009 CAF Champions League, tournament, Africa's premier club football tournament) (2009 CAF Champions League, tournament, football tournament)

Table 11: Additional examples of KG-Text datasets.

A.6. Cyclic Generation Examples

Input Text	Dataset	Predicted Graph	Reconstructed Text
Toba Domain was centered on Toba Castle in what is now the city of Toba.	LAGRANGE	(Toba Domain, capital, Toba) (Toba, instance of, city) (Toba Domain, headquarters location, Toba Castle)	Toba Domain was centered on Toba Castle in what is now the city of Toba.
	TeKGen	(Toba Domain, headquarters location, Toba Castle)	Toba Domain was centered on Toba Castle in what is now the city of Toba in Nagano Prefecture.
	T-REx	(Toba Castle, located in the administrative territorial entity, Toba)	Toba Castle (Serbian: \u0430 \u043c \u043e \u043a \u043e \u0430, Zamek Toba) is a castle in Toba, Serbo-Croatian Voivodeship, Poland.
	ChatGPT-GT	(Toba Domain, center, Toba Castle) (Toba Domain, location, city of Toba)	Toba Domain was centered on Toba Castle in the city of Toba.
	Guanaco-GT	(Toba Domain, centered on, Toba Castle) (Toba Domain, in, what is now the city of Toba) (Toba Domain, was, Toba Castle) (Toba Domain, was, what is now the city of Toba) (Toba Domain, was, centered on Toba Castle)	Toba Domain was centered on Toba Castle in what is now the city of Toba.
Mark Sandrich (born Mark Rex Goldstein; October 26, 1900 - 2013 March 4, 1945) was an American film director, writer, and producer.	LAGRANGE	(Mark Sandrich, date of death, March 4, 1945) (Mark Sandrich, occupation, film director) (Mark Sandrich, date of birth, October 26, 1900) (Mark Sandrich, occupation, film producer) (Mark Sandrich, date of death, 1945)	Mark Sandrich (October 26, 1900 - 2013 March 4, 1945), born Mark Sandritz, was an American film director and producer.
	TeKGen	(Mark Sandrich, date of death, 04 March 1945) (Mark Sandrich, occupation, Film director) (Mark Sandrich, date of birth, 26 October 1900)	Mark Sandrich (26 October 1900 - 4 March 1945) was a German film director.
	T-REx	(Mark Sandrich, country of citizenship, United States of America) (Mark Sandrich, occupation, film director)	Mark Sandrich is an American film director.
	ChatGPT-GT	(Mark Sandrich, occupation, film director) (Mark Sandrich, occupation, writer) (Mark Sandrich, occupation, producer) (Mark Sandrich, birth name, Mark Rex Goldstein) (Mark Sandrich, nationality, American)	Mark Sandrich (born Mark Rex Goldstein; July 7, 1939) is an American film director, writer, and producer.
	Guanaco-GT	(Mark Sandrich, born, Mark Rex Goldstein) (Mark Sandrich, died, March 4 1945) (Mark Sandrich, profession, film director) (Mark Sandrich, profession, writer) (Mark Sandrich, profession, producer) (Mark Sandrich, birth date, October 26 1900)	Mark Sandrich (born Mark Rex Goldstein; October 26, 1900 - 2013 March 4, 1945) was an American film director, writer, and producer.

Table 12: Generations of TGT cyclic evaluation.

A.7. Evaluation Prompts

We use the following prompt for measuring the quality of constructed datasets by asking GPT4 to rate the equivalence between a set of triples and the corresponding generated sentence.

[Instruction]

Please act as an impartial judge and evaluate the comparison between an original set of triples and a generated sentence produced from those triples. Instead of using discrete labels like "Equivalent," "Extra Information," or "Missing Information," you should provide a score between -10 to +10 for each case. -10 means extreme missing information, +10 means extreme hallucination, 0 means equivalence (either through logical deduction or literally). In your evaluation, consider factors such as relevance, accuracy, and completeness of the generated sentence. Assign a score based on the extent to which the generated sentence is equivalent to the original set of triples, contains extra information, or misses information.

As an example, if you believe that the generated sentence misses some information, you can assign a negative score (e.g., -3), and if you believe it contains some additional information, you can assign a positive score (e.g., +4). If the generated sentence both misses and contains extra information, assign the score that corresponds to the greater deviation from equivalence. For instance, if you assign -3 for missing information and +4 for extra information, the final score is +4 because 4 > 3.

[Examples]

Example 1:

[Original Set of Triples]
<S> Jack Smith <P> place of birth <O> Toronto <sep> <S> Jack Smith <P> occupation <O> engineer

[Generated Sentence]
Jack Smith is a Canadian engineer.

[Explanation]
The generated sentence logically deduces that Jack Smith is Canadian based on the information that he was born in New Westminster, which is in Canada. This logical deduction is consistent with the original set of triples, and thus, the sentence is equivalent.

[Relationship Score between Original Set of Triples and Generated Sentence]
0

Example 2:

[Original Set of Triples]
<S> Jack Smith <P> given name <O> Jack <sep> <S> Jack Smith <P> family name <O> Smith <sep> <S> Jack Smith <P> languages spoken, written, or signed <O> English <sep> <S> Jack Smith <P> occupation <O> engineer

[Generated Sentence]
Jack Smith is an English engineer.

[Explanation]
The generated sentence logically deduces that Jack Smith is an English engineer based on the provided given name, family name, language spoken, and occupation. This logical deduction is consistent with the original set of triples, and thus, the sentence is equivalent.

[Relationship Score between Original Set of Triples and Generated Sentence]
0

Example 3:

[Original Set of Triples]
<S> Olegarius<P> date of birth<O> 1060 <sep> <S> Olegarius <P> date of death <O> 1137 <sep> <S> Olegarius <P> position held <O> Bishop of Barcelona <sep> <S> Olegarius<P>work location <O> Tarragona <sep> <S> Olegarius <P> languages spoken, written or signed <O> Spanish

[Generated Sentence]
Olegarius was a Bishop of Barcelona, born in 1060, worked in Tarragona, and primarily spoke, wrote, and signed in Spanish. He passed away in 1137.

[Explanation]
The generated sentence conveys the same information as the original set of triples, providing details about Olegarius's date of birth, date of death, position held, work location, and languages spoken. It does not add or omit any significant information compared to the original set of triples.

[Relationship Score between Original Set of Triples and Generated Sentence]
0

Example 4:

[Original Set of Triples]
<S> Queensland <P> located in the administrative territorial entity <O> Australia <sep> <S> Toowoomba Region <P> located in the administrative territorial entity <O> Queensland <sep> <S> Harlaxton <P> located in the administrative territorial entity <O> Queensland

[Generated Sentence]
Queensland is located in Australia. Harlaxton is located in Queensland.

[Explanation]
The generated sentence omits information about Toowoomba Region being located in Queensland, which was present in the original set of triples.

[Relationship Score between Original Set of Triples and Generated Sentence]
-3

Example 5:

[Original Set of Triples]
<S> Mona Lisa <P> created by <O> Leonardo da Vinci <sep> <S> Mona Lisa <P> creation date <O> 1503 <sep> <S> Mona Lisa <P> location <O> Louvre Museum

[Generated Sentence]
The Mona Lisa, created by Leonardo da Vinci in 1503, is displayed at the Louvre Museum. It is widely regarded as one of the most famous art pieces globally.

[Explanation]
The generated sentence adds additional information about the Mona Lisa, including its significance as one of the most famous art pieces globally. This detail was not present in the original set of triples.

[Relationship Score between Original Set of Triples and Generated Sentence]
3

Example 6:

[Original Set of Triples]
<S> The Eiffel Tower <P> location <O> Paris, France <sep> <S> The Eiffel Tower <P> height <O> 330 meters <sep> <S> The Eiffel Tower <P> architect <O> Gustave Eiffel

[Generated Sentence]
Paris is a beautiful city in Europe.

[Explanation]
The generated sentence provides no information related to The Eiffel Tower, including its location, height, or architect, making it a complete miss of information.

[Relationship Score between Original Set of Triples and Generated Sentence]
-10

Example 7:

[Original Set of Triples]
<S> Albert Einstein <P> place of birth <O> Ulm, Germany <sep> <S> Albert Einstein <P> nationality <O> German <sep> <S> Albert Einstein <P> famous for <O> theory of relativity

[Generated Sentence]
Albert Einstein, the famous astronaut, was born on the moon and is known for discovering the secret to time travel.

[Explanation]
The generated sentence contains information that is entirely fabricated and has no connection to the original set of triples. It not only fails to match the original information but also introduces completely false details, making it a case of complete hallucination.

[Relationship Score between Original Set of Triples and Generated Sentence]
+10

[Original Set of Triples]
{original_set_of_triples}

[Generated Sentence]
{generated_sentence}

[Explanation]
[Provide your explanation here.]

[Relationship Score between Original Set of Triples and Generated Sentence]
[Assign a score between -10 and +10 based on the criteria mentioned in the instruction.]

In addition, we use the following prompt for measuring the quality of generated sentences in our cyclic evaluation. In particular, we ask GPT4 to rank generated sentences based on their equivalence to the original sentence.

[Instruction]

Please act as an impartial judge and rank the following three generated sentences based on their equivalence to the original sentence. Determine which of the generated sentences contains the least amount of missing information or hallucination and should be ranked first. Your ranking should be based on factors such as relevance, accuracy, and completeness. Be as objective as

possible.

After providing your explanation for each generated sentence, assign a rank from 1 to 3, with 1 being the highest (most equivalent) and 3 being the lowest (least equivalent) based on the criteria mentioned above.

[Examples]

Example 1: Ranking

[Original Sentence]
"The cat is on the mat."

[Generated Sentence 1]
"The black cat is on the mat."

[Explanation]
The generated sentence adds the detail "black" to describe the cat, which was not present in the original sentence.

[Rank]
2

[Generated Sentence 2]
"The cat is on the mat."

[Explanation]
The generated sentence is equivalent to the original sentence, containing no extra or missing information.

[Rank]
1

[Generated Sentence 3]
"The cat is on the mat outside."

[Explanation]
The generated sentence adds the detail "outside," which was not present in the original sentence.

[Rank]
3

[Original Sentence]
{original_sentence}

[Generated Sentence 1]
{generated_sentence_1}

[Explanation]
Evaluate the equivalence, relevance, and completeness of the generated sentence compared to the original sentence. Mention any missing information or additions.

[Rank]
[Assign a rank from 1 to 3]

[Generated Sentence 2]

{generated_sentence_2}

[Explanation]
Evaluate the equivalence, relevance, and completeness of the generated sentence compared to the original sentence. Mention any missing information or additions.

[Rank]
[Assign a rank from 1 to 3]

[Generated Sentence 3]
{generated_sentence_3}

[Explanation]
Evaluate the equivalence, relevance, and completeness of the generated sentence compared to the original sentence. Mention any missing information or additions.

[Rank]
[Assign a rank from 1 to 3]