# On Reducing Factual Hallucinations in Graph-to-Text Generation using Large Language Models

**Dmitrii Iarosh**[1, 3] **Alexander Panchenko**[1, 2] **Mikhail Salnikov**[2, 1]
[1]Skoltech [2]AIRI [3]Saint Petersburg State University
{D.Yarosh, A.Panchenko, Mikhail.Salnikov}@skol.tech

## Abstract

Recent work in Graph-to-Text generation has achieved impressive results, but it still suffers from hallucinations in some cases, despite extensive pretraining stages and various methods for working with graph data. While the commonly used metrics for evaluating the quality of Graph-to-Text models show almost perfect results, it makes it challenging to compare different approaches. This paper demonstrates the challenges of recent Graph-to-Text systems in terms of hallucinations and proposes a simple yet effective approach to using a general LLM, which has shown state-of-the-art results and reduced the number of factual hallucinations. We provide step-by-step instructions on how to develop prompts for language models and a detailed analysis of potential factual errors in the generated text.

## 1 Introduction

Knowledge Graphs KG have become a powerful tool for organizing and representing complex data due to their ability to easily manage trusted information and are used in various industries such as education, healthcare, and social media (Peng et al., 2023). They can be used in conjunction with modern Large Language Models (LLMs) to create Retrieval-Augmented Generation (RAG) systems or to validate the information generated or retrieved. Due to the better validation and understanding of data, it is important to properly translate them into text. This process, known as Graph-to-Text generation, has recently seen success in creating knowledge-grounded chatbots (Zhou et al., 2018; Peng et al., 2024) and Question Answering systems (Razzhigaev et al., 2023; Agarwal et al., 2021; Salnikov et al., 2023). Belikova et al. (2024) demonstrated that integrating various external resources, particularly linearized subgraphs, and employing a marginal probability-based selection method significantly enhanced the effectiveness of the RAG setup.

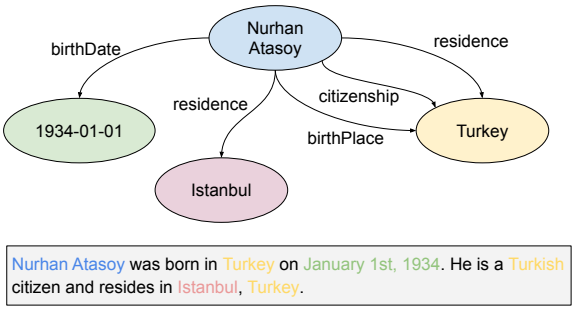



Figure 1: An example of a knowledge graph and its corresponding Graph-to-Text generation that describes the entities and their relationships in the provided graph.

Graph-to-Text involves the processing of Knowledge Graph triplets (subject, property, object) data, into a natural textual representation that should include all the factual information from these triplets and nothing else, as shown in the Figure 1.

In this work, we focus on evaluating the abilities of modern Large Language Models, such as ChatGPT[1], LLaMA-3 (Dubey et al., 2024) and Gemma 2 (Rivière et al., 2024), to solve the Graph-to-Text problem, and specifically on the potential hallucinations that are totally unacceptable for such problems.

Recent studies have produced state-of-the-art results in the Graph-to-Text generation task. They used complex pipelines to organize graphs in a specific order to generate correct text (Guo et al., 2020), or used graph aware approaches (Colas et al., 2022). All of these methods required complex training stages, but they provided the best results according to the leaderboards[2].

Despite the impressive results, these Graph-to-Text methods can still experience hallucinations and may omit certain elements of the graph in the

[1]https://openai.com/chatgpt/overview/
[2]https://synalp.gitlabpages.inria.fr/webnlg-challenge/challenge_2020



*Proceedings of the Generative AI and Knowledge Graph Workshop (GenAIK)*, pages 43–53
January 19, 2025. 

resulting text which are difficult to detect with the common metrics used on popular leaderboards. In this paper, we focus on this issue and provide a detailed analysis of state-of-the-art methods, comparing them with general Language Models.

Our contribution are two-fold:

- We provide a detailed guide to our prompt engineering strategy for LLMs in the Graph-to-Text domain, which allows users to achieve state-of-the-art results using a general LLM without the need for complex setup stages, fine-tuning, etc.
- We evaluate state-of-the-art methods and modern large language models (LLMs) on the popular Graph-to-Text dataset, WebNLG (Gardent et al., 2017b), and provide a new and detailed analysis to estimate the hallucinations of these methods which showed limitations of various Natural Language Generation (NLG) metrics.

We make our code publically available to provide reproducible results and motivate future researchers.[3]

## 2 Related work

In this section, we will provide a brief overview of Knowledge Graphs, their representation and linearization, as well as existing Graph-to-Text algorithms and the potential use of Large Language Models for this task.

### 2.1 Knowledge Graphs

Knowledge Graphs, such as Wikidata (Vrandečić and Krötzsch, 2014) or DBPedia (Lehmann et al., 2015), represent knowledge about their entities in a structured format. Connection between entities are labelled with properties and together they form triples (subject, property, object). Each triple describes a single fact about its entities without any unnecessary information. This type of knowledge organization helps to provide a brief description of each KG node by summarizing its neighbours. Additionally, Knowledge Graphs are easy to edit in order to keep the data in them up-to-date. There are two main ways to use Knowledge Graphs in conjunction with Language Models. The first one is to feed the graph directly into specially pretrained Graph Convolution Network and then use graph encoded version as input for language model decoder (Zhao et al., 2020). This approach save all the information about the graph structure but requires training a custom model. Other way is to linearize the graph in the special order: it can be connected with graph traversal (Ribeiro et al., 2020; Li et al., 2021) or selected by custom neural network (Guo et al., 2020). Such approach can lose information about two or more steps connections because of unstructured information representation in the Large Language Models context, but can be used with general purpose Large Language Model.

### 2.2 Graph-to-Text

Problem of Graph-to-Text generation started from lexicalisation task — converting individual Knowledge Graph triples into verb phrase templates. In the first presented solution text was generated based on the predefined templates (Goldberg et al., 1994). This algorithm is simple but requires custom human created templates for each task and can be applied only in a few specific areas. Future development in this area led to creation of algorithms for autonomous extraction of such templates from training data source (Duma and Klein, 2013; Perera and Nand, 2015). While such algorithms provide first fully autonomous generation pipeline, quality was still not enough to compare it with human-written results. Next step in this topic was done by WEBNLG dataset (Gardent et al., 2017a) for Graph-to-Text and vice versa generation. This dataset was based on DBpedia (Lehmann et al., 2015) and provides enough training data for ability to apply fine-tuned pretrained data-driven ML models for Graph-to-Text task. One of the first solutions based on this data was an LSTM transformer based model for sequence to sequence translation (Gardent et al., 2017b). Other approach used graph convolution network (GCN) as encoder which can process graph without linearization to save it structure (Marcheggiani and Perez-Beltrachini, 2018). Further develop of this solution led to the usage of graph attention networks as an encoder to provide more modelling power and improved performance (Koncel-Kedziorski et al., 2019). Another close solution (Beck et al., 2018) where graph-based encoder is also used, replaces GCN with Gated Graph Neural Network (Li et al., 2015). With active development of pretrained language models (PLMs) it was shown that is it possible to use graph embeddings by graph neural network as the input word embeddings of PLM for gener-

[3] https://github.com/s-nlp/llm-g2t

ating text after their representation alignment (Li et al., 2021). Relation-biased breadth first search was used to linearize the graph structure for PLM sequence decoder as it saves information about nodes at the same level with relevant semantics and forces more human-like order of description for relations. Another solution (Guo et al., 2020) proposes a relational graph convolutional network which is used as a planner to linearize the graph in the correct order before feeding it to the pretrained T5 model (Raffel et al., 2020) for the final text generation. This solution shows the best quality in the WEBNLG 2020 challenge. PLM can provide high quality result for the task of Graph-to-Text generation even without special graph-based neural networks (Ribeiro et al., 2020). Fine-tuned for a few epochs BART (Lewis et al., 2020) and T5 models were evaluated on the WEBNLG and AGENDA (Koncel-Kedziorski et al., 2019) datasets with linear traversal graph linearization. While such models as T5 and BART required extra fine-tuning on the task dataset modern general purpose Large Language Models like ChatGPT can provide comparable quality results for the Graph-to-Text translation even in zero-shot mode (Axelsson and Skantze, 2023). Comparison of ChatGPT and GPT-3 (Brown et al., 2020) Large Language Models with pretrained T5 and BART models shows that Large Language Models provide comparable quality results, but tend to generate text with hallucinations and irrelevant information (Yuan and Färber, 2023). In this work we will show that this problem can be solved by prompt engineering and replacing Large Language Model by modern one.

### 2.3 Large Language Models

Large language models such as Chat-GPT and GPT-3 are based on the transformer decoder architecture (Vaswani et al., 2017). They were designed to provide even zero-shot text generation for user request based on the huge train dataset and large amount of tuned parameters used in their training process. Development of these models by OpenAI lead to the next generation model called GPT-4 [4]. It demonstrates better quality on creative and long context tasks but it is also not open-source and OpenAI doesn't publish any paper about its architecture and training process. An alternative to the GPT-4 model was provided by Meta AI with their Llama 3 models family (Dubey et al., 2024). This Large Language Model is open-source and enough powerful to be used instead of GPT-4, according to provided results of comparison. At the same time Gemma 2 model (Rivière et al., 2024) was introduced by Google. It beats Llama 3 models of the nearly same size and even can be competitive among models with larger amount of parameters. It was also published to open-source.

Quality of Large Language Model answers can be enriched not only by applying new training techniques and increasing of train dataset or model parameters amount but also by different prompting techniques during evaluation of the user request. Providing Large Language Model with a few examples of processing the requested task can lead to the better performance and model adaptation to the new kinds of tasks (Brown et al., 2020). Such method is also known as few-shot prompting. Large Language Model are different from people in their process of thinking, so it is important to generate intermediate thoughts to provide better final quality of generation (Wei et al., 2022). It is called Chain-of-Thoughts method and can be applied both to zero-shot prompt using "Think step by step" phrase or even to a few-shot prompt by including examples with intermediate steps.

### 2.4 Fact Verification Metric

Employing modern Large Language Models for tasks like text summarization or graph-to-text translation produces favorable results; nevertheless, these models still have the propensity to hallucinate, and such hallucinations can be particularly harmful when they arise in factual statements. To detect factual inconsistency in the Large Language Models output factual consistency metrics such as AlignScore (Zha et al., 2023) can be used. AlignScore is based on RoBERTa (Liu et al., 2019) model which was trained to estimate the information alignment score between two arbitrary text pieces: context and claim. Given text pieces *context* and *claim*, *claim* is aligned with *context* if *context* contains all information from *claim* and *claim* does not contradict *context*. AlignScore was trained on several fact verification datasets (Schuster et al., 2021; Nie et al., 2019) that consist of claims paired with relevant contexts derived from Wikipedia [5] pages, alongside labels indicating the veracity of the claims. To enhance the metric's ability for continuous prediction, semantic text similar-

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

[5] https://www.wikipedia.org

ity datasets (Marelli et al., 2014; Cer et al., 2017) were incorporated into AlignScore's training corpus. These datasets consist of sentence pairs and corresponding similarity scores, illustrating the degree of semantic relatedness or independence between the sentences.

## 3 Proposed Framework

We propose a universal and easy-to-use framework for the Graph-to-Text task that does not require fine-tuning or the use of specialized trained modules or models, yet still achieves state-of-the-art results. We use common instruction-based Large Language Models, such as LLaMA 3 (Dubey et al., 2024), Gemma 2 (Rivière et al., 2024) or GPT-4o[6], to generate comprehensive, natural-style text from KG triplets using carefully selected prompt with various prompting techniques.

In our work, as with any system development, we start from a simple zero-shot baseline and simply ask the LLM to convert KG triples into text using the following prompt: *"Translate from graph to text"*. This straightforward approach suffers from a lot of hallucinations, so we asked the model not to hallucinate: *"Describe all nodes of the graph with edges as a connected text. Talk only about items from graph and use information only if graph contains it. Write only description without headers and titles."*, and it actually works, with better results. After that, we tried adding some general hacks, such as a few short learning examples and a chain of thoughts, to improve it. Finally, we provided the following prompt template:

Act as a system which describes all nodes of the graph with edges as a connected text. Follow the examples. Talk only about items from graph and use information only if graph contains it. Validate each written fact and correct it if mistake is found, do it silently without extra notes. Let's think step by step. For each step show described triple and check that all words from it is used in your description.

Task:

Graph: LINEARIZED GRAPH FROM EXAMPLE 1

Model answer:

Step-by-step solution:

MODEL STEP BY STEP SOLUTION FROM EXAMPLE 1

Description: MODEL GRAPH DESCRIPTION FROM EXAMPLE 1

... (Here comes more examples) ...

Now provide answer for the next task yourself.

Task:

Graph: LINEARIZED GRAPH

Model answer:

Step-by-step solution:

It actually works efficiently with the small LLaMA-3-8B-Instruct model and the big proprietary GPT-4o model. Examples of generated outputs for small graph, consisting of two triples (Figure 2), are presented in Table 1.

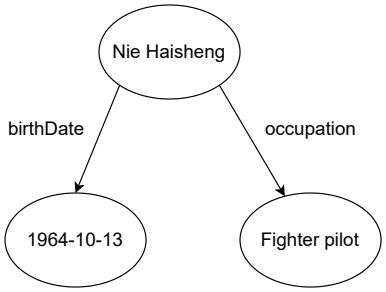



Figure 2: Visualization of the graph described in the Table 1

## 4 Experiments

Opposite to the previous works (Axelsson and Skantze, 2023; Yuan and Färber, 2023) for the evaluation we have used more actual Large Language Models such as open-source LLaMA-3-8B-Instruct, gemma-2-9b and proprietary gpt-4o and gpt-4o-mini. As a source of Knowledge Graphs we have chosen WEBNLG 2020 English dataset as it is one of the main benchmarks for the Graph-to-Text task. All models were evaluated on the test set using modified tripleset — graph description in the triple form where some predicates are replaced according to the official WEBNLG recommendation[7]. It helps models better understand directions of the links between entities. Each graph was linearized in the order triples are stated in the dataset. Example of the final graph input format is [graph][head] Expect a Miracle [relation] runtime [tail] 3274.0</s>, where [graph] and </s> denote start and end of the graph description and [head], [relation] and [tail] divide parts of the triple. We also replaced all underscore symbols by spaces to prevent Large Language Model from generation incorrect object names.

[6] https://openai.com/index/gpt-4

[7] https://synalp.gitlabpages.inria.fr/webnlg-challenge/docs/#triple-modification

| G2T Model | Prompt | BLEU↑ | BERTScore F1↑ | Example |
|---|---|---|---|---|
| Reference | – | 1 | 1 | Nie Haisheng, born on October 13, 1964, worked as a fighter pilot. |
| Llama-3-8B-Instruct | zero-shot (simple) | 0.1402 | 0.9455 | Nie Haisheng is a person born on October 13, 1964. As a Fighter pilot, he is a professional aviator. |
| Llama-3-8B-Instruct | zero-shot (anti-hallucinations) | 0.5548 | 0.9771 | Nie Haisheng, born on October 13, 1964, is a Fighter pilot. |
| Llama-3-8B-Instruct | few-shot | 0.2647 | 0.9754 | Nie Haisheng is a fighter pilot born on October 13, 1964. |
| Llama-3-8B-Instruct | CoT few-shot | 0.2370 | 0.9458 | The fighter pilot Nie Haisheng was born on October 13, 1964. |
| Gemma-2-9b | zero-shot (simple) | < 0.0001 | 0.9124 | Nie Haisheng's birthdate is 1964-10-13. Nie Haisheng's occupation is Fighter pilot. |
| Gemma-2-9b | zero-shot (anti-hallucinations) | < 0.0001 | 0.9238 | Nie Haisheng was born on 1964-10-13. Nie Haisheng's occupation is Fighter pilot. |
| Gemma-2-9b | few-shot | < 0.0001 | 0.9541 | Nie Haisheng was born on 1964-10-13 and is a Fighter pilot. |
| Gemma-2-9b | CoT few-shot | < 0.0001 | 0.9235 | Fighter pilot Nie Haisheng was born on 1964-10-13. |
| GPT-4o | zero-shot (simple) | 0.3388 | 0.9673 | Nie Haisheng was born on October 13, 1964, and hisoccupation is a fighter pilot. |
| GPT-4o | zero-shot (anti-hallucinations) | 0.3388 | 0.9673 | Nie Haisheng was born on October 13, 1964, and hisoccupation is a fighter pilot. |
| GPT-4o | few-shot | 0.4572 | 0.9797 | Nie Haisheng was born on October 13, 1964, and works as a fighter pilot. |
| GPT-4o | CoT few-shot | **0.6407** | **0.9839** | Nie Haisheng, born on October 13, 1964, is a fighter pilot. |

Table 1: Examples of Large Language Model outputs with different prompts on the graph, consisting of two triples. Blue text means model hallucinations.

To measure the results we have used standard WEBNLG metrics:Meteor (Banerjee and Lavie, 2005), BLEU (Papineni et al., 2002), Chrf (Popovic, 2015), TER (Snover et al., 2006) and BertScore (Zhang et al., 2020). Moreover, we additionally computed AlignScore metric (Zha et al., 2023) to detect factual inconsistency in the model answers.

For comparison reasons we have also evaluated GAP (Colas et al., 2022) and calculated metrics for the $P^2$ model (Guo et al., 2020), which is top-1 solution from WEBNLG 2020 competition, based on the model outputs published by authors[8]. Results of our evaluation are presented in the Table 2.

While by some metrics we can easily define the better model, it can be seen that BERTScore F1 is nearly equal both for Large Language Models and $P^2$ and requires more detailed analysis.

## 5 Analytics

While by classical translation metrics Large Language Models are slightly worse that the $P^2$ model, it was expected as fine-tuned models were adopted for the style of reference answers during training on the train part of the dataset and these metrics reward word match (Axelsson and Skantze, 2023). On one hand it gives $P^2$ advantage, but on the other hand it can't be applied to another dataset without extra fine-tuning process, while Large Language Models can be evaluated just with other examples in few-shot part of the prompt. As difference between Large Language Models and $P^2$ by BERTScore F1 is at the margin of statistical error we go deeper and compared factual consistency of the generated results with AlignScore. While Large Language Models all as one show high score by this metric, $P^2$ demonstrates much worse quality. It can be explained by hallucinations or missed facts in the model answers. To define the reasons of such problems with factual consistency we reviewed examples from the dataset where $P^2$ suffers from fact inconsistency, but two best of compared Large Language Models (Gemma 2 and GPT-4o) still provide high-quality results. One pattern we detected is that $P^2$ model tends to hallucinate if graph contains multiple triples with the same subject and property but different objects. Examples of the such graph

[8] https://github.com/QipengGuo/P2_WebNLG2020/blob/main/output.txt

| G2T Model | Setup | AlignScore↑ | Meteor↑ | BLEU↑ | Chrf↑ | TER↓ | BERTScore F1↑ |
|---|---|---|---|---|---|---|---|
| GAP | task-specific | 0.7797 | 0.5333 | 0.2398 | 0.5985 | 70.4437 | 0.9298 |
| $P^2$ | task-specific | 0.1511 | **0.6286** | **0.4054** | 0.6434 | **44.2396** | **0.9549** |
| Llama-3-8B-Instruct | zero-shot | 0.8959 | 0.5507 | 0.2690 | 0.6312 | 66.7051 | 0.9381 |
| Gemma-2-9b | zero-shot | 0.9100 | 0.5816 | 0.3148 | 0.6363 | 55.4666 | 0.9448 |
| GPT-4o | zero-shot | 0.8909 | 0.5970 | 0.2872 | **0.6559** | 69.0469 | 0.9455 |
| GPT-4o-mini | zero-shot | 0.8826 | 0.5940 | 0.2916 | 0.6488 | 66.8438 | 0.9442 |
| Llama-3-8B-Instruct | CoT few-shot | 0.9021 | 0.5487 | 0.2492 | 0.6136 | 62.5371 | 0.9432 |
| Gemma-2-9b | CoT few-shot | 0.9459 | 0.5818 | 0.3298 | 0.6300 | 48.2942 | 0.9517 |
| GPT-4o | CoT few-shot | **0.9514** | 0.6079 | 0.3402 | 0.6536 | 52.8860 | 0.9520 |
| GPT-4o-mini | CoT few-shot | 0.9436 | 0.5873 | 0.3036 | 0.6417 | 53.9540 | 0.9509 |

Table 2: Comparison of modern Large Language Models Graph-to-Text evaluation on WEBNLG 2020 dataset using simple zero-shot prompts and CoT few-shot prompts which also ask model not to hallucinate; AlignScore (Roberta-base).

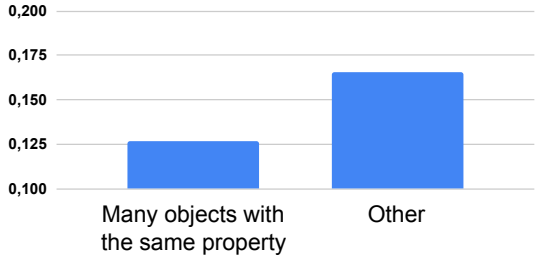



Figure 3: Comparison of AlignScore of $P^2$ model on graphs which contains multiple triples with the same subject and property but different objects and graphs without such triples.

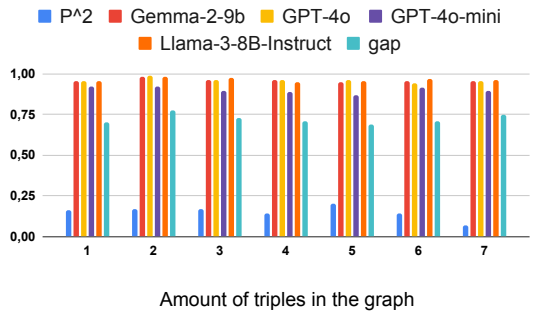



Figure 4: Comparison of AlignScore for $P^2$ and CoT few-shot prompted Large Language Models grouped by the graphs size in triples.

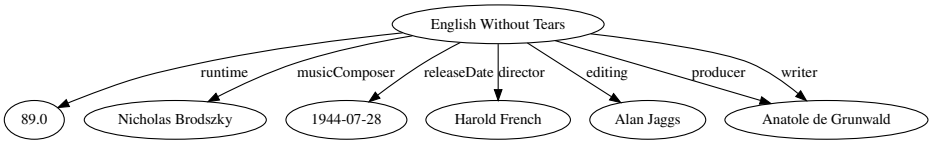
are provided in the Table 4. To prove this point we aggregated AlignScore results by cases where the graph satisfies this condition and cases where such triplets are absent. The comparison is shown in the Figure 3. It can be seen that $P^2$ shows 24% less quality in such situations.

Another problem is connected with the size of the graph provided to the model. While Large Language Models show stable quality on any number of triples in the graph $P^2$ loses more than 50% of quality on the graphs with seven triples. Example of such graph and models output are presented in the Table 3. Comparison of AlignScore for graphs with different triples count is provided in th Figure 4. We have also detected that even on smaller graphs $P^2$ often skips one of the facts from the graph which led to great but not full description. Examples are given in Table 3 and 5.

To sum up, $P^2$ shows great results by classic translation metrics because of special graph reordering and language model fine-tuning which makes model answer similar to the references, but still suffers from hallucinations more than modern Large Language Models because of the under the hood T5 model limitations. While generating the result with $P^2$ requires less computational resources it can be used in further processing only after factual consistency check to detect possible skipped or incorrect facts.

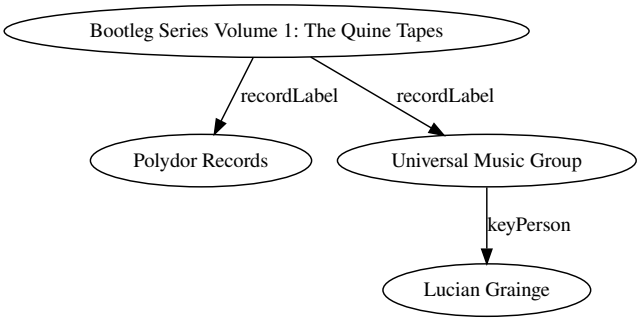
## 6 Conclusion

We have shown how to select prompt for the modern Large Language Model to show competitive results in Graph-to-Text task with top-1 specialized model fine-tuned for the specific dataset. We provide final few-shot prompt based on the Chain-of-Thoughts technique. Also we evaluated four modern Large Language Models on the graph-to-text task and compared their results with state-of-the-art solution in this task. Based on these evaluation we provided detailed analytics about potential factual errors which can be caused by model hallucinations. Moreover, we described how they can be omitted behind classic translation metrics on the example of the $P^2$ model.



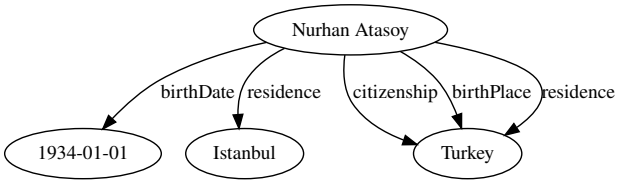

| | |
|---|---|
| **Reference** | English Without Tears is a 89 minute film that was written and produced by Anatole de Grunwald. The movie was released on July 28th, 1944. Nicholas Brodszky was the musical composer and Harold French was the director. The editing was done by Alan Jaggs. |
| $P^2$ | English Without Tears was produced by Anatole de Grunwald and written by Nicholas Brodszky. It was edited by Alan Jaggs and released on 28th July 1944. It has a runtime of 89.0 and was directed by Harold French. |
| **GPT-4o** | Released on July 28, 1944, "English Without Tears" runs for 89 minutes. The film was directed by Harold French and edited by Alan Jaggs. The music was composed by Nicholas Brodszky. Anatole de Grunwald contributed as both the producer and the writer. |
| **Gemma2** | *English Without Tears*, released on 1944-07-28, is a film with a runtime of 89.0 minutes. It was directed by Harold French and written by Anatole de Grunwald who also produced it. The music was composed by Nicholas Brodszky and edited by Alan Jaggs. |

Table 3: Example of the graph with seven triples where $P^2$ made a factual incorrect statement. Red text means incorrect fact detected by AlignScore, green text marks correct description




| | |
|---|---|
| **Reference** | Bootleg Series Volume 1: The Quine Tapes was released by Universal Music Group, a record label run by key person Lucian Grainge. Polydor Records recorded Bootleg Series Volume 1: The Quine Tapes. |
| $P^2$ | Lucian Grainge is a key person at Universal Music Group, the label that signed Bootleg Series Volume 1: The Quine Tapes to Polydor Records. |
| **GPT-4o** | Bootleg Series Volume 1: The Quine Tapes is associated with Polydor Records and Universal Music Group. A key person in Universal Music Group is Lucian Grainge. |
| **Gemma2** | Bootleg Series Volume 1: The Quine Tapes was released by both Polydor Records and Universal Music Group. Lucian Grainge is a significant figure at Universal Music Group. |

Table 4: Example of the graph with multiple triples with the same subject and property but different objects. Red text means incorrect fact detected by AlignScore, green text marks correct description




| | |
|---|---|
| **Reference** | Nurhan Atasoy was born in Turkey on January 1st, 1934. He is a Turkish citizen and resides in Istanbul, Turkey. |
| $P^2$ | Nurhan Atasoy was born in Turkey on January 1st 1934. He resides in Istanbul and has Turkish citizenship. |
| **GPT-4o** | Nurhan Atasoy, born in Turkey on January 1, 1934, holds Turkish citizenship and resides in Istanbul, Turkey. |
| **Gemma2** | Nurhan Atasoy, who was born on 1934-01-01 in Turkey, is Turkish and lives in Istanbul and Turkey. |

Table 5: Examples of graph to text generation with various models. The $P^2$ model omits one of the facts.

## 7 Limitations

The presented approach of Graph-to-Text translation using Large Language Models requires more computational resources than the state-of-the-art solution. Additionally, it is possible to measure the amount of model hallucinations using human evaluation in addition to the AlignScore.

## References

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.

Agnes Axelsson and Gabriel Skantze. 2023. Using large language models for zero-shot natural language generation from knowledge graphs. In *Proceedings of the Workshop on Multimodal, Multilingual Natural Language Generation and Multilingual WebNLG Challenge (MM-NLG 2023)*, pages 39–54, Prague, Czech Republic. Association for Computational Linguistics.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization@ACL 2005, Ann Arbor, Michigan, USA, June 29, 2005*, pages 65–72. Association for Computational Linguistics.

Daniel Beck, Gholamreza Haffari, and Trevor Cohn. 2018. Graph-to-sequence learning using gated graph neural networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 273–283, Melbourne, Australia. Association for Computational Linguistics.

Julia Belikova, Evegeniy Beliakin, and Vasily Konovalov. 2024. JellyBell at TextGraphs-17 shared task: Fusing large language models with external knowledge for enhanced question answering. In *Proceedings of TextGraphs-17: Graph-based Methods for Natural Language Processing*, pages 154–160, Bangkok, Thailand. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Daniel M. Cer, Mona T. Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity - multilingual and cross-lingual focused evaluation. *CoRR*, abs/1708.00055.

Anthony M. Colas, Mehrdad Alvandipour, and Daisy Zhe Wang. 2022. GAP: A graph-aware language model framework for knowledge graph-to-text generation. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 5755–5769. International Committee on Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Daniel Duma and Ewan Klein. 2013. Generating natural language from linked data: Unsupervised template extraction. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013) – Long Papers*, pages 83–94, Potsdam, Germany. Association for Computational Linguistics.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017a. 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.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017b. The WebNLG challenge: Generating text from RDF data. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 124–133, Santiago de Compostela, Spain. Association for Computational Linguistics.

E. Goldberg, N. Driedger, and R.I. Kittredge. 1994. Using natural-language processing to produce weather forecasts. *IEEE Expert*, 9(2):45–53.

Qipeng Guo, Zhijing Jin, Ning Dai, Xipeng Qiu, Xiangyang Xue, David Wipf, and Zheng Zhang. 2020. $P^2$: A plan-and-pretrain approach for knowledge graph-to-text generation. In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 100–106, Dublin, Ireland (Virtual). Association for Computational Linguistics.

Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, and Hannaneh Hajishirzi. 2019. Text Generation from Knowledge Graphs with Graph Transformers. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2284–2293, Minneapolis, Minnesota. Association for Computational Linguistics.

Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, and Hannaneh Hajishirzi. 2019. Text generation from knowledge graphs with graph transformers. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2284–2293. Association for Computational Linguistics.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, S. Auer, and Christian Bizer. 2015. Dbpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6:167–195.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. 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, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics.

Junyi Li, Tianyi Tang, Wayne Xin Zhao, Zhicheng Wei, Nicholas Jing Yuan, and Ji-Rong Wen. 2021. Few-shot knowledge graph-to-text generation with pretrained language models. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 1558–1568. Association for Computational Linguistics.

Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard S. Zemel. 2015. Gated graph sequence neural networks. *arXiv: Learning*.

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. *CoRR*, abs/1907.11692.

Diego Marcheggiani and Laura Perez-Beltrachini. 2018. Deep graph convolutional encoders for structured data to text generation. In *Proceedings of the 11th*

*International Conference on Natural Language Generation*, pages 1–9, Tilburg University, The Netherlands. Association for Computational Linguistics.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014*, pages 216–223. European Language Resources Association (ELRA).

Yixin Nie, Haonan Chen, and Mohit Bansal. 2019. Combining fact extraction and verification with neural semantic matching networks. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6859–6866. AAAI Press.

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, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL.

Boci Peng, Yun Zhu, Yongchao Liu, Xiaohe Bo, Haizhou Shi, Chuntao Hong, Yan Zhang, and Siliang Tang. 2024. Graph retrieval-augmented generation: A survey. *CoRR*, abs/2408.08921.

Ciyuan Peng, Feng Xia, Mehdi Naseriparsa, and Francesco Osborne. 2023. Knowledge graphs: Opportunities and challenges. *Artif. Intell. Rev.*, 56(11):13071–13102.

Rivindu Perera and Parma Nand. 2015. A multi-strategy approach for lexicalizing linked open data. In *Computational Linguistics and Intelligent Text Processing*, pages 348–363, Cham. Springer International Publishing.

Maja Popovic. 2015. chrf: character n-gram f-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation, WMT@EMNLP 2015, 17-18 September 2015, Lisbon, Portugal*, pages 392–395. The Association for Computer Linguistics.

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. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Anton Razzhigaev, Mikhail Salnikov, Valentin Malykh, Pavel Braslavski, and Alexander Panchenko. 2023. A system for answering simple questions in multiple languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 524–537, Toronto, Canada. Association for Computational Linguistics.

Leonardo F. R. Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. 2020. Investigating pretrained language models for graph-to-text generation. *CoRR*, abs/2007.08426.

Morgane Rivière, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozinska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucinska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju-yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjösund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, and Lilly McNealus. 2024. Gemma 2: Improving open language models at a practical size. *CoRR*, abs/2408.00118.

Mikhail Salnikov, Hai Le, Prateek Rajput, Irina Nikishina, Pavel Braslavski, Valentin Malykh, and Alexander Panchenko. 2023. Large language models meet knowledge graphs to answer factoid questions. In *Proceedings of the 37th Pacific Asia Conference on Language, Information and Computation*, pages 635–644, Hong Kong, China. Association for Computational Linguistics.

Tal Schuster, Adam Fisch, and Regina Barzilay. 2021. Get your vitamin c! robust fact verification with contrastive evidence. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 624–643. Association for Computational Linguistics.

Matthew G. Snover, Bonnie J. Dorr, Richard M. Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted

human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers, AMTA 2006, Cambridge, Massachusetts, USA, August 8-12, 2006*, pages 223–231. Association for Machine Translation in the Americas.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Shuzhou Yuan and Michael Färber. 2023. Evaluating generative models for graph-to-text generation. In *Recent Advances in Natural Language Processing*.

Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. 2023. Alignscore: Evaluating factual consistency with A unified alignment function. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 11328–11348. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Chao Zhao, Marilyn Walker, and Snigdha Chaturvedi. 2020. Bridging the structural gap between encoding and decoding for data-to-text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2481–2491, Online. Association for Computational Linguistics.

Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. 2018. Commonsense knowledge aware conversation generation with graph attention. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 4623–4629. ijcai.org.