Enhancing Graph-to-Text Systems in Low-Resource Settings: Distilling Chain-Of-Thought Reasoning For Task-Specific Workflows

David Guzman Piedrahita
22-737-571, nethz: dguzman
dguzman@ethz.ch

Arnisa Fazla
22-740-039, nethz: afazla
afazla@ethz.ch

Anna Kiepura
21-744-032, nethz: akiepura
akiepura@student.ethz.ch

Abstract

In this study, we investigate the efficacy and performance of graph-to-text systems in low-resource settings. We initially reproduce and extend previous work by integrating a more diverse dataset and an augmented dataset.

In our initial experiment, we performed multi-task training for the graph-to-text task on a single dataset and assessed its performance on others with different input structures. An additional model was then trained on a merged dataset (WebNLG, DART and E2E) and subsequently evaluated on each dataset individually. Despite the varying nature of the datasets, we found comparable performance across all models, including the ones trained on just one dataset, which we attribute to our standardized dataset preprocessing.

Our second experiment focuses on enhancing graph-to-text systems via knowledge distillation. We leverage the capabilities of ChatGPT to augment a restricted training dataset, subsequently training smaller T5 flan models (small and base variants) on it.

Our findings illustrate that a pipeline approach that emulates task-specific chain-of-thought reasoning can outperform traditional end-to-end models when the parameter count and number of training samples is limited, suggesting the possibility of achieving performance gains associated with larger models in smaller, less resource-intensive ones.

1 Introduction

This study explores the task of data-to-text generation, specifically focusing on graph-to-text translation with an emphasis on knowledge graphs. We delve into various techniques previously proposed in literature, investigating their efficacy in improving Large Language Model (LLM) performance on this task. This is accomplished through the application of recent LLM concepts, including chain-of-thought reasoning (Wei et al., 2023a) and multi-task learning (Wei et al., 2022), with no adjustments to model architecture. The experimental setup is conducted under low-resource conditions, employing multiple datasets to generate a heterogeneous set, and incorporating external LLMs like ChatGPT (Radford et al., 2018) to augment the dataset. The enhancement is performed by introducing an intermediate 'Reasoning' step between input triples and graph descriptions.

Two key findings emerged from our investigation. First, preprocessing was identified as a significant factor in improving knowledge transfer efficiency in multi-task training or knowledge transfer contexts for smaller models (details in section 5.2). Second, we found that the chain-of-thought property, usually found in massive LLMs (Wei et al., 2023a), could be simulated in smaller models, for task-specific scenarios, by introducing an intermediate task-specific reasoning objective.

We argue that the introduction of an intermediate reasoning step can potentially improve graph-to-text task performance. Benefits include enhanced coverage of input graph information, and better interpretability (Wei et al., 2023b). Despite these promising findings, future research is required to confirm these benefits and explore other possible applications of splitting the graph-to-text task.

The corresponding code for reproducing our experiments is available in our Github repository.

2 Related Work

The task of graph-to-text generation aims to create coherent texts from graph data by comprehending the information the graph presents. Early strategies utilized text-to-text neural models, which leveraged sequences of Knowledge Graph (KG) triples as input and natural language as output (Konstas et al., 2017; Trisedya et al., 2018; Moryossef et al., 2019a;

Castro Ferreira et al., 2019a). Alternatively, certain methodologies concentrated on encoding the structural data from graphs, employing tools designed for graph data processing such as Graph Convolutions (Kipf and Welling, 2017; Ribeiro et al., 2019; Damonte and Cohen, 2019; Ribeiro et al., 2020) and Graph Neural Networks (Gori et al., 2005; Koncel-Kedziorski et al., 2022), which extended attention mechanisms for direct graph input training.

Further advanced transformer-based methodologies incorporated structural graph data into transformer architectures. For instance, Ribeiro et al. (2020) and Zhang et al. (2020) integrated graph attention in their encoders to incorporate graph structures directly. Additionally, Li et al. (2022) enhanced overall performance and computational efficiency by selectively pruning graph components deemed unnecessary or redundant for text generation.

Modern Pretrained Language Models (PLMs), such as Bert (Devlin et al., 2019) and T5 (Raffel et al., 2020), have significantly advanced baselines for various natural language understanding tasks. Researchers have subsequently strived to utilize these PLMs in graph-to-text generation, representing graph structures as input strings to produce superior text output (Mager et al., 2020; Radev et al., 2020; Clive et al., 2022). Utilizing task adaptive pre-training (Gururangan et al., 2020) and introducing Control Prefixes, which expands upon prefix-tuning (Li and Liang, 2021), have further enhanced performance (Clive et al., 2022).

Despite these advancements, studies have highlighted the limitations of linearizing graph structures within PLMs, indicating that this could damage the input graph's connectivity information (Song et al., 2018; Beck et al., 2018; Ke et al., 2021; Ribeiro et al., 2021b). Furthermore, the dissimilarity between the structured graph input and the natural language the PLMs are trained on hinders full knowledge transfer when fine-tuning (Ke et al., 2021). To address these issues, research has developed Structure-aware semantic aggregation modules and proposed pre-training strategies to better align graph and text embedding spaces (Ke et al., 2021). These approaches and others, such as the enhancement of graph structure-aware trainable weights (Ribeiro et al., 2021b) and the improvement of adapters (Colas et al., 2022), have made strides in preserving graph structure and improving encoding during the PLM stage.

While the field of graph-to-text generation has made significant progress, challenges remain in graph scalability, interpretability, and oversmoothing (Wei et al., 2023b). The potential of Large Language Models (LLMs) to improve these areas has been highlighted, with Wei et al. proposing that certain properties of LLMs only emerge once a certain parameter threshold is met, including the ability to transfer knowledge between different tasks or domains, multi-step reasoning and instruction following. Furthermore, recent research into knowledge distillation offers potential pathways for transferring knowledge from large to smaller language models where inference time is a concern (Eldan and Li, 2023; Gunasekar et al., 2023; Mukherjee et al., 2023).

3 Experiment Settings

3.1 Datasets

Our project utilizes various datasets that have been specifically developed for this type of task. One of these datasets, which represents our primary source of training data, is WebNLG (Moryossef et al., 2019b). Its the corpus consists of collections of triplets that depict information about entities and the relationships between them, accompanied by corresponding natural language text. Each set in the corpus contains a maximum of 7 triplets.

Following the interim analysis, we decide to focus mostly on the WebNLG dataset. This decision was influenced by our computational resources constraint and the desire for reliable intra-experiment comparison of results. Furthermore, due to chatGPT API limitations 4.3, we extracted a subset of 1000 samples from the WebNLG's training data to be used for the chain-of-thought experiments, creating a low-resource training environment .

The knowledge transfer in multi-task training experiment, however, by definition did require additional datasets namely DART and E2E.

The DART dataset (Nan et al., 2021) comprises a vast array of structured tables and their corresponding natural language descriptions. It stands out for its wide domain coverage and the complexity of the relations depicted in the tables. Its size and domain diversity make it particularly suited for the knowledge transfer in multi-task training experiment.

Another dataset that we have incorporated in our work is the E2E (Novikova et al., 2017). The E2E, short for ”End-to-End,” dataset is specifically designed for the task of natural language generation from structured data. The dataset comprises a collection of structured representations describing various aspects of a restaurant (such as its food, area, and price range) along with the corresponding natural language descriptions. This dataset has a more narrowly focused domain compared to DART and WebNLG, but its strength lies in its large number of unique ways of describing similar entities and properties, providing a rich resource for the generation of diverse and nuanced language.

For all the datasets, we prepare the input graphs by linearizing the triples to a string form and removing irregularities such as the samples with emmpty target texts before fine-tuning. For WebNLG We first add new special tokens to the tokenizer to represent the elements of the triples in the graph: '$\langle H \rangle$' for the head entity, '$\langle R \rangle$' for the relation, and '$\langle T \rangle$' for the tail entity. We preprocess DART and E2E similarly, (see 4.2).

3.2 Evaluation Metrics

In evaluating our model, we employ BLEU (Post, 2018) and CHR F++ scores (Popović, 2017) as our primary metrics, adopting these from Ribeiro et al. (2021a), whose experiment we replicate. These metrics, computed using a script from https://github.com/WebNLG/GenerationEval, provide insights into how well our model's generated texts align with the reference texts within the validation set. The BLEU score, which measures the overlap of predicted n-grams with reference n-grams. The CHR F++ score extends this assessment to character-level overlaps.

Despite their utility, these metrics alone cannot fully encapsulate semantic equivalence, hence we supplement them with BERT-Scores (Zhang et al., 2019), better suited to capture semantic correlation, along with a qualitative evaluation to assess the correspondence of translations.

3.3 Models

We tested FLAN-T5 (small and base) in our experiments to evaluate its performance in normal and low-resource situations, particularly assessing its behavior when trained on augmentation data (see 4.3).

FLAN-T5 differs from traditional T5, used in (Ribeiro et al., 2021a), in its fine-tuning approach. While traditional T5 uses task-specific fine-tuning, FLAN-T5 (Chung et al., 2022) uses instruction finetuning, making it versatile and efficient for unseen tasks.

Moreover, to the best of our knowledge, this is the first study to use FLAN-T5-small in graph-to-text generation, traditionally done using T5 and similar models.

4 Experimental Findings

In this section we describe our experiments which investigate how we can utilize LLMs for the graph-to-text task, where the input graph is preprocessed to be a linearized string of texts, so the model architecture does not need to be changed. We experimented with various techniques to improve the performance of graph-to-text conversion using large language models.

First, we reproduced (Ribeiro et al., 2021a) from scratch, where we used task-adaptive pretraining and finetuning to adapt the model to graph-to-text tasks.

Second, we leveraged the power of multi-task learning by introduced tasks that shared a similar underlying structure with the graph-to-text task, to examine whether this would lead to an improvement in performance compared to the baseline models.

Lastly, we implemented an intermediate reasoning step, emulating the chain-of-thought property found in massive language models. This step aimed to break down the task of graph-to-text conversion into more manageable subtasks, and thus facilitating a more controlled processing of information.

Our results not only demonstrate the efficacy of these methods but also provide insight into their operation and the aspects that contribute most to their success. In the previous section we discuss the experimental setup, datasets, metrics and the models trained, and this section will describe the results obtained.

4.1 Finetuning FLAN-T5 small for graph to text

In this subsection we discuss how we utilized a pretrained Transformer model, specifically FLAN-T5-small, and fine-tuned it on the WebNLG dataset for this purpose. Our experiments assessed the model's performance and adaptability to this specific task.

Model preparation: task adaptive pretraining and finetuning This workflow is a reproduction of (Ribeiro et al., 2021a). Although the original code was available on GitHub, we re-implemented the entire process due to conflicting dependencies and incomplete parts of the script, which could also contribute to the difference in results.

Prior to fine-tuning, we pretrained the T5 flan small model in a masked (15%) language model task using the texts available in the WebNLG dataset. The objective was to enable the model to get a better understanding of the linguistic patterns in the dataset.

After pretraining, we prepared the dataset for the main task: graph-to-text generation. We processed the dataset to form pairs of input-output sentences where the input is the graph represented as triples and the output is the corresponding text. We trained the model on these pairs for five epochs, with a noticeable reduction in training loss across epochs.

4.1.1 Automatic Evaluation

We evaluated the performance of our fine-tuned FLAN-T5-small model using the 2017 edition of the WebNLG dataset. The evaluation metrics used were BLEU (Bilingual Evaluation Understudy), NLTK's BLEU (BLEU score calculated using the Natural Language Toolkit), and chrF++, which is an automatic evaluation metric based on character n-gram precision, recall, and f-score.

The model achieved a BLEU score of 51.41. This score indicates that the generated text had a substantial level of overlap with the reference text in terms of n-grams. However, the BLEU score calculated using NLTK was 0.5. The difference in scores might be attributed to the different calculation methods used by these two tools.

For the chrF++ score, which calculates the character n-gram precision, recall, and f-score, the model achieved a score of 0.65. This demonstrates that the generated text had a high degree of similarity with the reference text at the character level.

Overall, the results obtained indicate that the FLAN-T5-small model, when fine-tuned on the WebNLG dataset, produced satisfactory performance in the graph-to-text generation task close to the performance achieved by (Ribeiro et al., 2021a). We attribute the small discrepancies between the original paper and our reproduction to our more limited hyperparameter tuning caused by compute and time constraints.

4.2 Multi-Task Learning

In an effort to harness the potential of multi-task learning, we incorporated supplementary tasks that bear a similar fundamental structure to our primary graph-to-text task. The intention behind this strategy was to determine whether this incorporation could bolster the model's performance relative to the baseline models.

Custom data pre-processing for Multi-Task Learning In our pursuit of efficient Multi-Task Learning, we devised Python classes to cater to each dataset's unique preprocessing requirements. The classes, namely WebNLGDataset, DartDataset, E2ENLGDataset, and BalancedCombinedDataset, are designed to process WebNLG, Dart, and E2ENLG datasets respectively.

The WebNLGDataset class parses and converts the input graph triples into textual format, deploying custom tags `<H>`, `<R>`, and `<T>` for delimitation. The translation task from graph to text is introduced with an appropriate prompt (”translate from graph to text”). Similarly, the DartDataset class, with a few necessary adjustments, applies the same preprocessing approach to the Dart dataset.

The E2ENLGDataset class, tasked with translating meaning representations to text, introduces a distinct prompt to guide the model's context (”translate from meaning representation to text”) and gives the linearized meaning representation as an input. The BalancedCombinedDataset class is used for amalgamating WebNLG, Dart, and E2ENLG datasets. It addresses possible training bias by ensuring exposure to an equal count of instances from each dataset. If the datasets vary in length, the shorter ones are appropriately resized to match the largest dataset.

These classes provide an effective data preprocessing strategy for multi-task learning, incorporating context-specific instruction and structure to enhance task-specific learning.

4.2.1 Automatic Evaluation

From the metrics in table 1, table 2 and table 3, we can conclude that our multi-task learning approach generally demonstrated comparable, and in some cases slightly superior, performance across different datasets and metrics when compared to a model trained solely on a single task. In fact, the models trained exclusively on WebNLG and DART respectively show competent performance across all three datasets, which is particularly interesting in the case of E2E, given that the input format is completely different from the one used in WebNLG and DART. For further analysis refer to 5.2 .

Test Dataset	BERT precision	BERT recall	BERT f1	BLEU	chrF++
WebNLG 2020	0.937	0.931	0.934	38.96	0.59
DART	0.946	0.942	0.944	36.49	0.62
E2E	0.942	0.944	0.943	30.84	0.59

Table 1: The metric scores of the T5 flan base model trained on **all three datasets**, evaluated on the three datasets separately. It can be seen that the model performances are close evaluated in all three datasets, which is unexpected considering there is a size difference between them.

Test Dataset	BERT precision	BERT recall	BERT f1	BLEU	chrF++
WebNLG 2020	0.937	0.932	0.934	38.71	0.59
DART	0.945	0.942	0.943	35.04	0.62
E2E	0.941	0.944	0.943	28.62	0.59

Table 2: The metric scores of the T5 flan base model trained on the **WebNLG 2020** dataset, evaluated on the three datasets separately.

4.2.2 Qualitative Analysis

In figures 2 and 3, you can see the predictions of the T5 flan base model trained on the 2020 WebNLG dataset and T5 flan base model trained on the DART dataset. For both samples, it can be seen that both models did not skip any information contained in the input structure, and only the model trained on WebNLG made a mistake (in ”Chinese coffee shop”) in figure 3. In fact, the reference texts given in figure 2 is missing parts of the information contained in the input data, for example it does not mention the rating of the coffee shop, which means predictions of both models have better quality than the reference for that sample.

4.3 LLM Data Augmentation

Generating a chain of thought (a series of intermediate reasoning steps) was found to improve the ability of Large Language Models (LLMs) to perform complex reasoning in a previous study by Wei et al. (2023a). The authors showed how such reasoning abilities emerge naturally in sufficiently large language models via a method called chain-of-thought prompting. This method involves guiding the model's responses by adding an intermediate reasoning step to the reference outputs during LLMs fine-tuning. Chain-of-thought prompting encourages the model to think step by step, as opposed to coming to the final output text right away.

The authors of the original study showed that chain-of-thought prompting significantly improves the performance on a range of arithmetic, commonsense, and symbolic reasoning tasks. However, the authors found that the chain-of-thought capability emerges from approximately 100B parameters onwards. To our knowledge, it was never explored if utilizing this method can also improve the performance of LLMs for the graph-to-text generation task. Moreover, we found no studies that attempted to mimic this technique on smaller LLMs, such as T5 small (60M parameters) and T5 base (220M parameters). Therefore, we drew inspiration from this study and knowledge distillation techniques (Gunasekar et al. (2023), Mukherjee et al. (2023)) to infuse chain-of-thought reasoning into T5 small and base models on the WebNLG dataset for graph-to-text generation. In addition, following the findings of Castro Ferreira et al. (2019b) that using a task-specific pipelined architecture improves the quality of the generated text in graph-to-text task, we give a qualitative comparison of the texts generated by the pipelined model and the end-to-end model.

4.3.1 Technical Details

To construct a modified version of the dataset that contains the intermediate reasoning step, which consists of transforming each triple into an independent sentence, we prompted ChatGPT with a prompt with a two-shot example. We used this prompt to generate the intermediate reasoning steps for 1000 training samples from the the WebNLG dataset. Subsequently, we split the generated samples into two datasets for training two separate models in our pipeline; 1. triples-to-reasoning and 2. reasoning-to-text.

Inspired by Castro Ferreira et al. (2019b) and given the small sizes of the T5 flan small and base models, we used two separate instances in a pipeline: one for generating the interme-

Test Dataset	BERT precision	BERT recall	BERT f1	BLEU	chrF++
WebNLG 2020	0.934	0.927	0.93	36.34	0.57
DART	0.945	0.94	0.942	34.71	0.61
E2E	0.943	0.944	0.944	30.52	0.6

Table 3: The metric scores of the T5 flan base model trained on the **DART** dataset, evaluated on the three datasets separately.

diate reasoning steps based on our augmented dataset (triples-to-reasoning), and a second one that takes these generated intermediate reasoning steps as input, and outputs the traditional target text (reasoning-to-text).

4.3.2 Automatic Evaluation

The evaluation results from the test dataset are summarized in table 4. The findings suggest that the use of an augmented dataset, with an intermediate reasoning step, enabled us to achieve good results in a low-resource situation, despite doubling the number of parameters by using two T5 instances in a sequential setup. T5 small struggled to generalize the knowledge gained from the augmented dataset, whereas T5 base demonstrated this ability. The superior performance of T5 base over T5 small could be attributed to its larger parameter count (see section 5.3 for more details).

4.3.3 Qualitative Analysis

To examine the qualitative properties of the generated text such as graph coverage and fluency, we conduct qualitative analysis of individual output samples. For most samples, we observe that both pipeline and single models can capture the same amount of information (i.e. they are usually able to translate the same triples). Confirming the findings of (Castro Ferreira et al., 2019b) that the end-to-end models generated text might have hallucinations (extra information in the output text which does not exist in the input text), in sample 5 from the appendix we see that behaviour in the end-to-end model generated text. The hallucinated part is: ”61.0 people per square mile”, which, while related to the concept of ”population density”, it still is adding extra information which does not exist in the input text. Also, in both 4 and 6, it can be seen that the end-to-end model repeats some information, which hurts the fluency of the generated text. The format of the intermediate reasoning output of the pipeline model can be seen in 6.

5 Analysis and Discussion

In this part we give a detailed analysis of our findings from section 4.

5.1 Finetuning FLAN-T5 small for graph to text

Qualitative Analysis of Model Predictions Across Epochs In addition to quantitative performance metrics -which are limited in scope in this context- it is important to understand how the model's performance evolves qualitatively across training epochs. For this purpose, we examine the model's predictions for the same test smaple over the course of five epochs.

This particular example taken from the test dataset, in Figure 1, refers to a graph describing the life and achievements of William Anders, an astronaut who served as a crew member of Apollo 8. The graph inputs for this example include several interconnected facts about Anders, such as his date and place of birth, his selection by NASA, his role in the Apollo program, and his retirement.

Overall, the progression of the model's performance across epochs appears to show some inconsistency. After an improvement from Epoch 1 to 2, the model seems to fluctuate in its ability to correctly capture all details. One observation is that the model never mentions Anders' place of birth or his correct association with Apollo 8. This could be an area for potential improvement.

This analysis suggests that while the model is indeed learning and improving over time, there might be a plateau effect where it starts to hover around a specific performance level. It may benefit from further fine-tuning, additional pretraining, or the integration of other training strategies to better understand and generate the subtleties of the graph-to-text generation task.

5.2 Multi-Task Learning

As seen in table 3, our model trained solely on the WebNLG dataset, the smallest dataset among the three (consisting of 13,211 training samples

Test Dataset	**BERT precision**	**BLEU**	**BLEU NLTK**	**chrF++**	**TER**
T5-flan-small Pipeline					
Triples-to-Reasoning	0.969	70.98	0.71	0.81	0.29
Reasoning-to-Text	0.929	46.53	0.46	0.68	0.68
Pipelined Model	0.923	41.81	0.41	0.65	0.72
*Pipelined Mode*l	0.897	19.98	0.2	0.5	0.87
T5-flan-small Single Model	0.945	52.46	0.52	0.65	0.49
T5-flan-small Single Model	0.916	28.3	0.28	0.54	0.75
T5-flan-base Pipeline					
Triples-to-Reasoning	0.970	67.55	0.68	0.78	-
Reasoning-to-Text	0.948	59.67	0.59	0.71	-
Pipelined Model	0.945	53.25	0.53	0.66	-
Pipelined Model	**0.935**	**37.46**	**0.37**	**0.6**	**0.57**
T5-flan-base Single Model	0.946	55.69	0.55	0.67	0.47
T5-flan-base Single Model	0.929	34.3	0.34	0.6	0.65

Table 4: The metric scores of the T5-flan (small and base) models, trained as a pipeline consisting of 2 models (triples-to-reasoning and reasoning-to-text) or trained as a single model. **Bold** models are either a pipeline consisting of a triples-to-reasoning and a reasoning-to-text model, or single end-to-end models. The rows with *Italic* models show the evaluation results of those models on the complete test set of the 2020 edition of the WebNLG, where rows that are not italic show the evaluation results on only 200 test samples containing the intermediate reasoning data, which is needed to evaluate the triples-to-reasoning and reasoning-to-text models separately. In the table, we only compare the scores which are on the italic rows, since the scores evaluated on the whole dataset are more accurate than the ones that were evaluated on only 200 samples.

compared to DART's 30,526 and E2E's 42,061), demonstrated remarkable generalization capability. Despite the considerable differences between the datasets, particularly between WebNLG's triple-based format and E2E's unique meaning representation structure, the model maintained comparable performance across all datasets.

This finding underscores the robustness and adaptability of transformer-based models such as T5, even when they are trained on smaller, task-specific datasets. The performances we observed seem to suggest that the preprocessing steps, specifically the standardization of input format across tasks, might play a crucial role in enabling models trained on a single task to perform commendably on the other tasks. In fact, in our case, we converted the three datasets into a shared format for WebNLG and DART, and adapted a similar format for E2E. This process likely eased the translation of learned knowledge between tasks, effectively mitigating the input format variations between datasets.

Conversely, the nearly identical performance between the multi-task model and the single-dataset models suggests that smaller models may not be well-suited for multi-task learning, as detailed in prior work(Wei et al., 2022). This limitation is thought to stem from the model's capacity to integrate and leverage learned patterns across disparate tasks. For this reason, it is important to consider the possibility that the model's size could be a limiting factor in this scenario. Specifically, the size of the model may restrict the potential performance boost facilitated by knowledge transfer in multi-task learning.

5.3 LLM Data Augmentation

As observed in Table 4, the performance of the pipeline model, which is based on the flan-T5-small, is not on par with the traditional end-to-end model across all measured metrics. However, a different trend emerges when the comparison is based on the flan-T5-base models; in this case, the pipeline model surpasses the performance of the end-to-end model.

Upon an in-depth analysis, it becomes apparent that both the flan-T5-small and flan-T5-base models exhibit comparable performance in the initial part of the pipeline, specifically the 'triples-to-reasoning' stage. In contrast, the latter section, 'reasoning-to-text,' presents a more substantial challenge for both model variations, creating a performance bottleneck when both sub-models are incorporated into the pipeline. Interestingly, this latter stage exhibits significant performance improvement in the case of the flan-T5-base model, thus elevating the potential performance ceiling. We propose that this enhanced upper boundary contributes to the pipeline model's ability to outperform the end-to-end model.

Furthermore, we anticipate that this trend could potentially extend to moderately larger models such as the flan-T5-large. However, it is unlikely to persist indefinitely. We hypothesize that significantly larger models, pretrained on extensive internet datasets and refined via reinforcement learning with human feedback, would likely surpass the flan-T5-base model in data-to-text tasks without requiring any fine-tuning or pipelining, given the considerable difference in their respective sizes.

This observed performance improvement in smaller models, facilitated through the pipeline approach, bears resemblance to the gains achieved via the 'chain of thought' reasoning strategy, typically associated with larger, around 100 billion parameter models (Wei et al., 2022). 'Chain of thought' reasoning, a method of structured interaction with the model over multiple steps, has been conventionally understood to become truly effective in the realm of these significantly larger models (Wei et al., 2023a). In our study, however, the pipeline model's performance enhancement may suggest that a semblance of this 'chain of thought' reasoning can be emulated in significantly smaller models such as the flan-T5-base. It's essential to clarify that we are not suggesting that T5 base possesses this full-fledged chain-of-thought reasoning. Rather, our results imply that task-specific chain-of-thought-style reasoning can be effectively incorporated or distilled into smaller models through a pipeline approach.

Consequently, this finding provides insight into more effective utilization of smaller models, potentially democratizing the benefits of 'chain of thought' reasoning via dataset augmentation, pipelining and finetuning.

6 Discussion and Conclusion

In our study, we addressed the challenges in existing graph-to-text systems in low-resource settings, building upon prior research (Ribeiro et al., 2021a; Wei et al., 2023b). Our methodology primarily comprised reproducing the aforementioned work and extending it with the approach of utilizing a more diverse or an augmented training dataset.

Our first major experiment, beyond the reproduction of (Ribeiro et al., 2021a; Wei et al., 2023b), involved multi-task training in the graph-to-text task, in which the model was trained on a singular dataset and subsequently evaluated on others with varying input structures. Additionally, we trained the model on a combined dataset encompassing all three of the datasets (WebNLG, DART and E2E) subsequently testing it on each dataset separately. Surprisingly, the performance of all models was comparable for all models - an unexpected result considering the datasets' diverse nature. We attribute this to our dataset preprocessing, which standardizes the input format across all datasets (see 3.1).

In the second experiment, our training dataset was restricted to 700 samples, we implemented an intermediate step to leverage the power of ChatGPT. This effectively divided the graph-to-text task into two subtasks: graph-to-reasoning (or triples-to-reasoning) and reasoning-to-text. We applied knowledge distillation (Ba and Caruana, 2014), utilizing ChatGPT to create a refined dataset to train a smaller model - specifically T5 flan (small and base variants). The objective was for the smaller model to approximate the larger model's predictions, enhancing the triples-to-reasoning model's performance. As evidenced in table 4, transitioning from T5-small to T5-base didn't influence the triples-to-reasoning model's performance but improved the reasoning-to-text model, thereby making the T5 flan base pipelined model superior to the baseline (T5 flan base single model).

Our study was limited by computational power. With more resources, we could experiment with larger models like T5-large, providing more insights into how our methodology performs with varying model sizes. Nevertheless, our aim remains to develop a compact model capable of delivering robust results, and concurrently studying the applicability of principles from massive language models to smaller models.

Encouragingly, our results illustrate the potential of preprocessing, with single dataset training proving effective in testing on other datasets. Additionally, we demonstrate the utility of partitioning tasks into subtasks in a pipeline, as a form to exploit the capabilities of larger LLMs to teach task-specific chain-of-thought reasoning to much smaller models.

References

Lei Jimmy Ba and Rich Caruana. 2014. Do deep nets really need to be deep?

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.

Thiago Castro Ferreira, Chris van der Lee, Emiel van Miltenburg, and Emiel Krahmer. 2019a. Neural data-to-text generation: A comparison between pipeline and end-to-end architectures. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 552–562, Hong Kong, China. Association for Computational Linguistics.

Thiago Castro Ferreira, Chris van der Lee, Emiel van Miltenburg, and Emiel Krahmer. 2019b. Neural data-to-text generation: A comparison between pipeline and end-to-end architectures. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 552–562, Hong Kong, China. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Jordan Clive, Kris Cao, and Marek Rei. 2022. Control prefixes for parameter-efficient text generation.

Anthony Colas, Mehrdad Alvandipour, and Daisy Zhe Wang. 2022. Gap: A graph-aware language model framework for knowledge graph-to-text generation. *arXiv preprint arXiv:2204.06674*.

Marco Damonte and Shay B. Cohen. 2019. Structural neural encoders for amr-to-text generation.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Ronen Eldan and Yuanzhi Li. 2023. Tinystories: How small can language models be and still speak coherent english?

M. Gori, G. Monfardini, and F. Scarselli. 2005. A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pages 729–734 vol. 2.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. 2023. Textbooks are all you need.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Pei Ke, Haozhe Ji, Yu Ran, Xin Cui, Liwei Wang, Linfeng Song, Xiaoyan Zhu, and Minlie Huang. 2021. JointGT: Graph-text joint representation learning for text generation from knowledge graphs. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2526–2538, Online. Association for Computational Linguistics.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks.

Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, and Hannaneh Hajishirzi. 2022. Text generation from knowledge graphs with graph transformers.

Ioannis Konstas, Srinivasan Iyer, Mark Yatskar, Yejin Choi, and Luke Zettlemoyer. 2017. Neural AMR: Sequence-to-sequence models for parsing and generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 146–157, Vancouver, Canada. Association for Computational Linguistics.

Liang Li, Ruiying Geng, Bowen Li, Can Ma, Yinliang Yue, Binhua Li, and Yongbin Li. 2022. Graph-to-text generation with dynamic structure pruning.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation.

Manuel Mager, Ramón Fernandez Astudillo, Tahira Naseem, Md Arafat Sultan, Young-Suk Lee, Radu Florian, and Salim Roukos. 2020. GPT-too: A language-model-first approach for AMR-to-text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1846–1852, Online. Association for Computational Linguistics.

Amit Moryossef, Yoav Goldberg, and Ido Dagan. 2019a. Step-by-step: Separating planning from realization in neural data-to-text generation. 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 2267–2277, Minneapolis, Minnesota. Association for Computational Linguistics.

Amit Moryossef, Yoav Goldberg, and Ido Dagan. 2019b. Step-by-step: Separating planning from realization in neural data-to-text generation. *CoRR*, abs/1904.03396.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive learning from complex explanation traces of gpt-4.

Linyong Nan, Dragomir Radev, Rui Zhang, Amrit Rau, Abhinand Sivaprasad, Chiachun Hsieh, Xiangru Tang, Aadit Vyas, Neha Verma, Pranav Krishna, Yangxiaokang Liu, Nadia Irwanto, Jessica Pan, Faiaz Rahman, Ahmad Zaidi, Mutethia Mutuma, Yasin Tarabar, Ankit Gupta, Tao Yu, Yi Chern Tan, Xi Victoria Lin, Caiming Xiong, Richard Socher, and Nazneen Fatema Rajani. 2021. Dart: Open-domain structured data record to text generation.

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2017. The E2E dataset: New challenges for end-to-end generation. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 201–206, Saarbrücken, Germany. Association for Computational Linguistics.

Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Dragomir R. Radev, Rui Zhang, Amrit Rau, Abhinand Sivaprasad, Chiachun Hsieh, Nazneen Fatema Rajani, Xiangru Tang, Aadit Vyas, Neha Verma, Pranav Krishna, Yangxiaokang Liu, Nadia Irwanto, Jessica Pan, Faiaz Rahman, Ahmad Zaidi, Murori Mutuma, Yasin Tarabar, Ankit Gupta, Tao Yu, Yi Chern Tan, Xi Victoria Lin, Caiming Xiong, and Richard Socher. 2020. DART: open-domain structured data record to text generation. *CoRR*, abs/2007.02871.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

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.

Leonardo F. R. Ribeiro, Claire Gardent, and Iryna Gurevych. 2019. Enhancing AMR-to-text generation with dual graph representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3183–3194, Hong Kong, China. Association for Computational Linguistics.

Leonardo F. R. Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. 2021a. Investigating pretrained language models for graph-to-text generation.

Leonardo F. R. Ribeiro, Yue Zhang, Claire Gardent, and Iryna Gurevych. 2020. Modeling global and local node contexts for text generation from knowledge graphs.

Leonardo F. R. Ribeiro, Yue Zhang, and Iryna Gurevych. 2021b. Structural adapters in pretrained language models for amr-to-text generation.

Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. 2018. A graph-to-sequence model for AMR-to-text generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1616–1626, Melbourne, Australia. Association for Computational Linguistics.

Bayu Distiawan Trisedya, Jianzhong Qi, Rui Zhang, and Wei Wang. 2018. GTR-LSTM: A triple encoder for sentence generation from RDF data. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1627–1637, Melbourne, Australia. Association for Computational Linguistics.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models. *Transactions on Machine Learning Research*. Survey Certification.

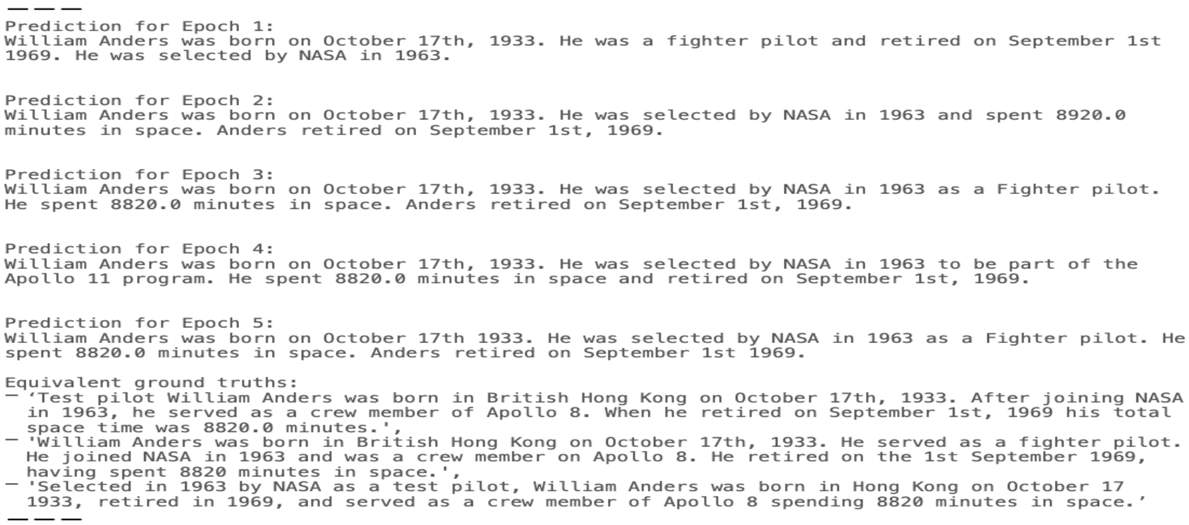
Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023a. Chain-of-thought prompting elicits reasoning in large language models.

Shaopeng Wei, Yu Zhao, Xingyan Chen, Qing Li, Fuzhen Zhuang, Ji Liu, and Gang Kou. 2023b. Graph learning and its applications: A holistic survey.

Jiawei Zhang, Haopeng Zhang, Congying Xia, and Li Sun. 2020. Graph-bert: Only attention is needed for learning graph representations.

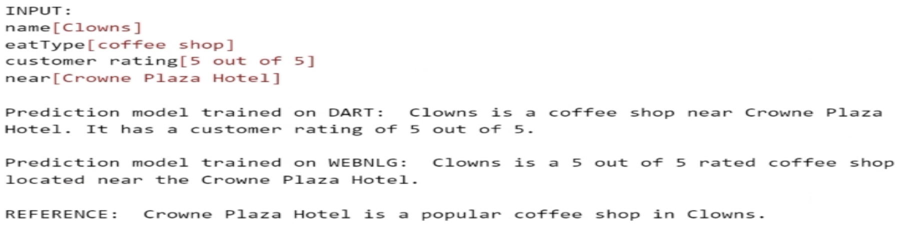
Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with BERT. *CoRR*, abs/1904.09675.

Appendix: Sample Predictions



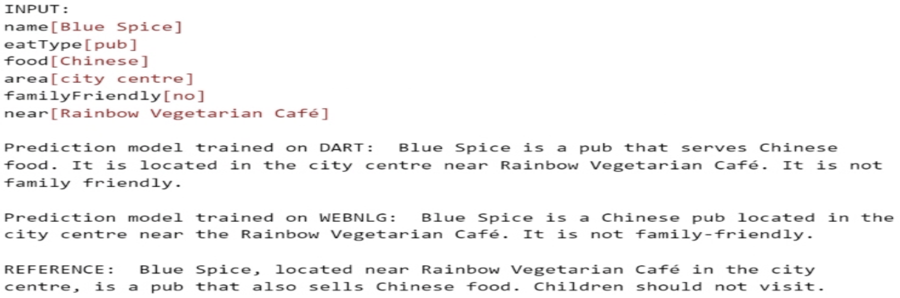
```
———
Prediction for Epoch 1:
William Anders was born on October 17th, 1933. He was a fighter pilot and retired on September 1st
1969. He was selected by NASA in 1963.

Prediction for Epoch 2:
William Anders was born on October 17th, 1933. He was selected by NASA in 1963 and spent 8920.0
minutes in space. Anders retired on September 1st, 1969.

Prediction for Epoch 3:
William Anders was born on October 17th, 1933. He was selected by NASA in 1963 as a Fighter pilot.
He spent 8820.0 minutes in space. Anders retired on September 1st, 1969.

Prediction for Epoch 4:
William Anders was born on October 17th, 1933. He was selected by NASA in 1963 to be part of the
Apollo 11 program. He spent 8820.0 minutes in space and retired on September 1st, 1969.

Prediction for Epoch 5:
William Anders was born on October 17th 1933. He was selected by NASA in 1963 as a Fighter pilot. He
spent 8820.0 minutes in space. Anders retired on September 1st 1969.

Equivalent ground truths:
— ‘Test pilot William Anders was born in British Hong Kong on October 17th, 1933. After joining NASA
  in 1963, he served as a crew member of Apollo 8. When he retired on September 1st, 1969 his total
  space time was 8820.0 minutes.',
— 'William Anders was born in British Hong Kong on October 17th, 1933. He served as a fighter pilot.
  He joined NASA in 1963 and was a crew member on Apollo 8. He retired on the 1st September 1969,
  having spent 8820 minutes in space.',
— 'Selected in 1963 by NASA as a test pilot, William Anders was born in Hong Kong on October 17
  1933, retired in 1969, and served as a crew member of Apollo 8 spending 8820 minutes in space.’
———
```

Figure 1: Progression in the model's performance, epoch per epoch, in one test sample



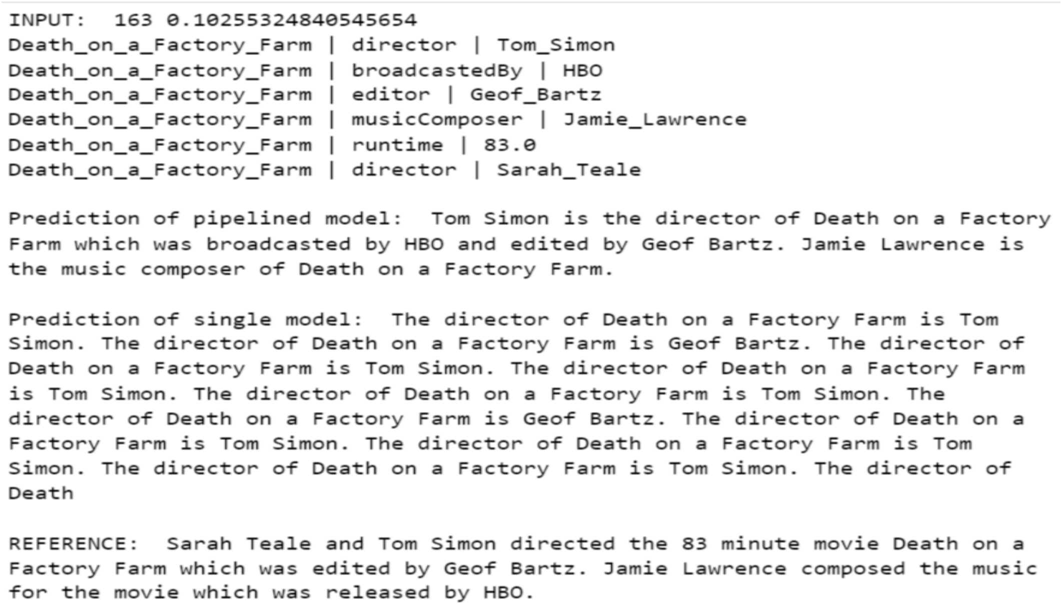
```
INPUT:
name[Clowns]
eatType[coffee shop]
customer rating[5 out of 5]
near[Crowne Plaza Hotel]

Prediction model trained on DART:  Clowns is a coffee shop near Crowne Plaza
Hotel. It has a customer rating of 5 out of 5.

Prediction model trained on WEBNLG:  Clowns is a 5 out of 5 rated coffee shop
located near the Crowne Plaza Hotel.

REFERENCE:  Crowne Plaza Hotel is a popular coffee shop in Clowns.
```

Figure 2: Predictions of models trained separately on WebNLG and DART datasets given the E2E dataset as an input



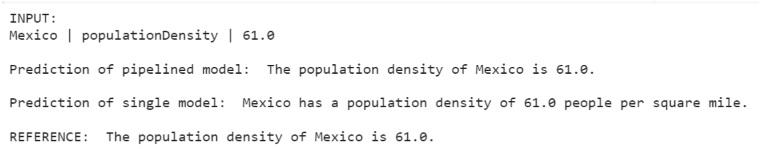
```
INPUT:
name[Blue Spice]
eatType[pub]
food[Chinese]
area[city centre]
familyFriendly[no]
near[Rainbow Vegetarian Café]

Prediction model trained on DART:  Blue Spice is a pub that serves Chinese
food. It is located in the city centre near Rainbow Vegetarian Café. It is not
family friendly.

Prediction model trained on WEBNLG:  Blue Spice is a Chinese pub located in the
city centre near the Rainbow Vegetarian Café. It is not family-friendly.

REFERENCE:  Blue Spice, located near Rainbow Vegetarian Café in the city
centre, is a pub that also sells Chinese food. Children should not visit.
```

Figure 3: Predictions of models trained separately on WebNLG and DART datasets given another E2E dataset sample as an input



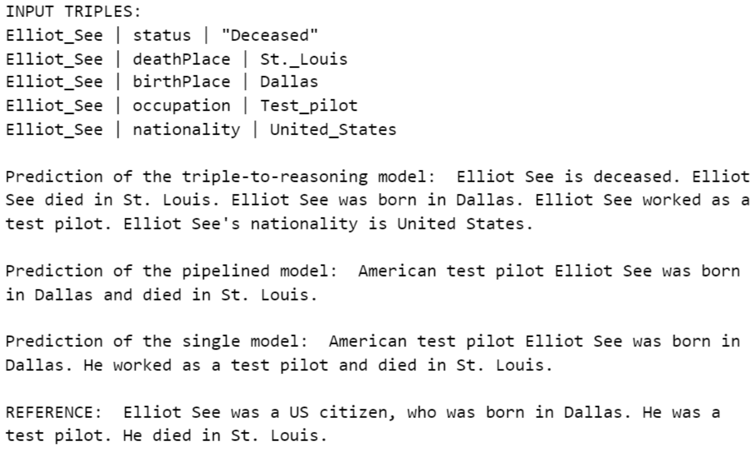
```
INPUT:  163 0.10255324840545654
Death_on_a_Factory_Farm | director | Tom_Simon
Death_on_a_Factory_Farm | broadcastedBy | HBO
Death_on_a_Factory_Farm | editor | Geof_Bartz
Death_on_a_Factory_Farm | musicComposer | Jamie_Lawrence
Death_on_a_Factory_Farm | runtime | 83.0
Death_on_a_Factory_Farm | director | Sarah_Teale

Prediction of pipelined model:  Tom Simon is the director of Death on a Factory
Farm which was broadcasted by HBO and edited by Geof Bartz. Jamie Lawrence is
the music composer of Death on a Factory Farm.

Prediction of single model:  The director of Death on a Factory Farm is Tom
Simon. The director of Death on a Factory Farm is Geof Bartz. The director of
Death on a Factory Farm is Tom Simon. The director of Death on a Factory Farm
is Tom Simon. The director of Death on a Factory Farm is Tom Simon. The
director of Death on a Factory Farm is Geof Bartz. The director of Death on a
Factory Farm is Tom Simon. The director of Death on a Factory Farm is Tom
Simon. The director of Death on a Factory Farm is Tom Simon. The director of
Death

REFERENCE:  Sarah Teale and Tom Simon directed the 83 minute movie Death on a
Factory Farm which was edited by Geof Bartz. Jamie Lawrence composed the music
for the movie which was released by HBO.
```

Figure 4: Predictions of the pipeline and single T5 flan base models given a sample from the 2020 edition of WebNLG dataset



```
INPUT:
Mexico | populationDensity | 61.0

Prediction of pipelined model:  The population density of Mexico is 61.0.

Prediction of single model:  Mexico has a population density of 61.0 people per square mile.

REFERENCE:  The population density of Mexico is 61.0.
```

Figure 5: Predictions of the pipeline and single T5 flan base models given a sample from the 2020 edition of WebNLG dataset



```
INPUT TRIPLES:
Elliot_See | status | "Deceased"
Elliot_See | deathPlace | St._Louis
Elliot_See | birthPlace | Dallas
Elliot_See | occupation | Test_pilot
Elliot_See | nationality | United_States

Prediction of the triple-to-reasoning model:  Elliot See is deceased. Elliot
See died in St. Louis. Elliot See was born in Dallas. Elliot See worked as a
test pilot. Elliot See's nationality is United States.

Prediction of the pipelined model:  American test pilot Elliot See was born
in Dallas and died in St. Louis.

Prediction of the single model:  American test pilot Elliot See was born in
Dallas. He worked as a test pilot and died in St. Louis.

REFERENCE:  Elliot See was a US citizen, who was born in Dallas. He was a
test pilot. He died in St. Louis.
```

Figure 6: Predictions by the single model and the pipeline model which has the *Reasoning* intermediate step given the input triples.