Retrieval-Augmented Modular Prompt Tuning for Low-Resource Data-to-Text Generation

Ruitao Feng, Xudong Hong, Mayank Jobanputra, Mattes Warning, Vera Demberg
Saarland University, Germany
{fruitao, xhong, mayank, mwarning, vera}@coli.uni-saarland.de

Abstract

Data-to-text (D2T) generation describes the task of verbalizing data, often given as attribute-value pairs. While this task is relevant for many different data domains beyond the traditionally well-explored tasks of weather forecasting, restaurant recommendations, and sports reporting, a major challenge to the applicability of data-to-text generation methods is typically data sparsity. For many applications, there is extremely little training data in terms of attribute-value inputs and target language outputs available for training a model. Given the sparse data setting, recently developed prompting methods seem most suitable for addressing D2T tasks since they do not require substantial amounts of training data, unlike finetuning approaches. However, prompt-based approaches are also challenging, as a) the design and search of prompts are non-trivial; and b) hallucination problems may occur because of the strong inductive bias of these models. In this paper, we propose a retrieval-augmented modular prompt tuning (RAMP) method, which constructs prompts that fit the input data closely, thereby bridging the domain gap between the large-scale language model and the structured input data. Experiments show that our RAMP method generates texts with few hallucinations and achieves state-of-the-art performance on a dataset for drone handover message generation.

Keywords: data-to-text generation, natural language generation, prompt tuning, retrieval augmentation

1. Introduction

In data-to-text (D2T) generation, the goal is to generate natural language descriptions such as y in Table 1 from structured data inputs x in Table 1. Recent methods leverage large-scale pretrained language models (PLMs) like GPT-3 (Brown et al., 2020) and T5 (Raffel et al., 2020) and perform in-context learning through *prompting* to adapt to different D2T tasks (Cao and Wang, 2022; Luo et al., 2022), showing impressive performance. In-context learning needs no or very few labelled instances and is therefore particularly attractive for low-resource tasks, as is the case for many D2T tasks which lack in-domain corpora.

Prompt-based generation exhibits one major challenge when employed on the low-resource D2T task: hallucinations are relatively frequent due to the strong inductive bias of PLMs (Ji et al., 2023; McKenna et al., 2023), especially when in-domain data is not readily part of the pretraining data (Keymanesh et al., 2022). For example, prompting the state-of-the-art PLM Vicuna-13B (Chiang et al., 2023) with the input data x in Table 1 generates a phrase *has collided with* (see y_0 in Table 1) that is contradicting the input data.

To tackle this challenge, one can employ *few-shot prompting* (Brown et al., 2020) by including examples in the prompt. However, the choice of examples is non-trivial because only examples that are similar to the input data are effective (Shuster et al., 2021).

To demonstrate the importance of prompt augmentation in low-resource D2T tasks, We propose a novel method named Retrieval-Augmented Modular Prompt Tuning (RAMP). We make the following contributions: **a)** We apply retrieval augmentation to select examples that are semantically similar to the input in terms of attributes in the input data; **b)** We employ modular prompts where we use the attributes to route the trainable continuous prompts into an augmented prompt x' such that it matches the input better; **c)** We conduct experiments on a low-resource D2T task with automatic and human evaluations to show the effectiveness of our approach.

2. Related Work

PLMs have demonstrated remarkable performance on D2T tasks. To enable prompting, PLMs are pretrained using curated texts {p, x, y} that are found in the web-content (Radford et al., 2019) or manually constructed (Sanh et al., 2022; Wei et al., 2022). At test time, PLMs are prompted to generate y given {p, x}. Zero-shot prompting only adds task-specific instruction to the prompt p, while few-shot prompting uses several input-output pairs from the training examples like {p, x_r, y_r, ..., x, y} and has repeatedly been found to outperform zero-shot prompting approaches (Brown et al., 2020).

However, the prompt-based approach raises the question of what an optimal prompt should look like. When some training data is available, it is possible to optimize the prompt using the training data. This is referred to as *prompt tuning*. It differs from fine-tuning in that it selectively optimizes specific

LREC-COLING 2024, pages 14053–14062
20-25 May, 2024.

Name	Notation	Example
Input data	x	{ "time_stamp": "0:05", "name": "castle", "Distance": 2.5 }
Gold output	y	The drone is facing the risk of physical damage. There is a castle in the drone's flight path at a distance of 2.5m
Prompt	p	Here is the raw sensor data from a drone: x. Write the handover messages that only includes crucial situation for this data.
Prompting output	y_0	The drone is facing the risk of physical damage. The drone has collided with a castle.
DL expression	E	[Distance : 2.5 $\sqsubseteq$ [Distance $\leq$ 3.0m] $\sqsubseteq$ **VeryClose**] [Time_stamp : "0:05" $\wedge$ Name : "castle" $\wedge$ **VeryClose**] $\sqsubseteq$ [VeryClose.Object] [$\exists$ VeryClose.Object] $\sqsubseteq$ RiskOfPhysical-Damage
Retrieved input	x_r	{ "time_stamp": "0:01", "name": "gravestone 1", "Distance": 3.0 }
Retrieved output	y_r	(0:01) The drone is facing the risk of physical damage. There is a gravestone in the drone's flight path at a distance of 3.0m.
Augmented prompt	x'	$[A_1, A_2..., p, x_r, y_r, x]$

Table 1: A sample data point of D2T task. The phrase has collided with is an example of intrinsic hallucination. The retrieved input-output pair (x_r, y_r) has the same attribute Distance with the input x. The continuous prompt A_1 is activated because it corresponds to the attribute Distance.

prompts, keeping the bulk of the PLM parameters frozen (Lester et al., 2021; Keymanesh et al., 2022). Furthermore, *modular prompt* (Chen et al., 2022b) uses a sequence of trainable prompts, each encoding knowledge related to a corresponding class of the data. Another way to create better prompts is to add semantically similar examples to them (Liu et al., 2022, 2023). Our approach leverages an attribute-based retriever to find relevant examples from structured data.

Our approach aims to combine the beneficial properties of different approaches – the fluency and grammaticality achieved by PLMs, with the reliability of few-shot prompt-based approaches, in a setting where very little data is available. To the best of our knowledge, our investigation represents the first exploration to combine retrieval augmentation and modular prompt for prompt augmentation.

2.1. Dataset

To demonstrate the effectiveness of our method on a low-resource dataset, we use a Drone dataset consisting of drone sensor data and handover messages to the human pilot in critical situations (Chang et al., 2022). This dataset only comprises 1654 data points, each manually annotated with realistic data records, capturing the dynamics of surrounding objects and 25 types of drone attributes such as altitude, flying speed, and battery level. Besides the full set of measurements, the dataset also contains a set of attribute-value pairs that are relevant to the handover situation (e.g., low battery or high winds). These critical attribute-value pairs represent the input to the D2T generation step. The input data has an average token count of 540.79, with token numbers ranging from 274 to 2481. The handover messages contain 148.54 tokens on average. The message lengths range from 29 tokens to 1263 tokens. We split the data into training, validation and test sets as described in Chang et al. (2022). A sample data point is shown in Table 1.

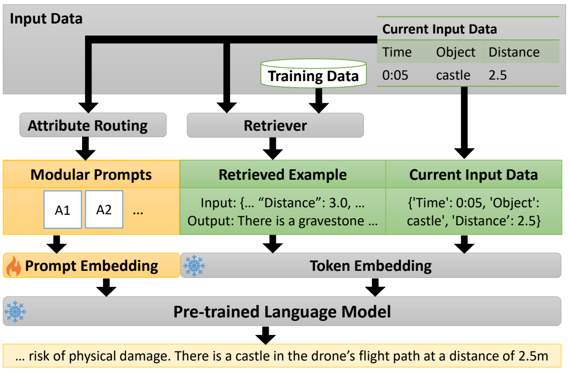
Input Data
Current Input Data
Time Object Distance
0:05 castle 2.5
Training Data
Attribute Routing
Retriever
Modular Prompts
A1
A2
...
Retrieved Example
Input: {... "Distance": 3.0, ...
Output: There is a gravestone ...
Current Input Data
{'Time': 0:05, 'Object': castle', 'Distance': 2.5}
Prompt Embedding
Token Embedding
Pre-trained Language Model
... risk of physical damage. There is a castle in the drone's flight path at a distance of 2.5m

Figure 1: The architecture of RAMP. It consists of an attribute-based router and an attribute-based retriever, continuous prompt tokens, and a frozen large language model. RAMP only requires training the modular prompt to align latent distributions of the data to the PLM.

3. Methodology

We aim to improve low-resource data-to-text generation by making the prompt optimally relevant to the input data. We achieve this by 1) retrieving the most similar examples from the training data and 2_a) learning a continuous prompt that optimally fits our task and 2_b) applying a modular method to replace continuous prompts with attribute-specific modular prompts. The overall architecture of our system is shown in Figure 1.

3.1. Retrieval Augmentation

We propose a Retrieval Augmentation method to retrieve training instances with similar activated logical expressions and combinations of attributes.

Firstly, we retrieve training instances with similar combinations of attributes. To do this, we obtain the activated logical expressions E from the current input data based on description logic defined in

Chang et al. (2022). For example, the description logic expression E in Table 6 describes a critical situation of a drone flying very close to an object. Then we construct a mapping from E to the most representative data point (x_r, y_r) in the training data that has similar attributes. All selected examples are in Appendix A.2. These retrieved examples are added to the input prompt, effectively serving as a template specifically for the input.

3.2. Modular Prompt Tuning

Prompt tuning We use a parameter-efficient transfer learning method named prompt tuning (Lester et al., 2021) for adapting PLMs to our domain and task. Prompt tuning freezes all the parameters of the PLM and incorporates a trainable continuous prompt, facilitating seamless integration of structured data into the model's input. The prompt embedding matrix is initialized randomly and trained end-to-end to generate context-aligned output texts. We use 20 continuous prompt tokens in our main experiments. We also experiment with other numbers and report the results in Appendix A.3.

Modular Prompt with Attribute Routing To augment the effectiveness of the continuous prompt, we incorporate modular prompts which use a separate prompt for different input data types (Chen et al., 2022b). We develop a collection of modular prompts, each aligned with distinct sets of attributes that underpin specific critical scenarios. This modular framework helps in injecting domain-specific knowledge, allowing PLMs to comprehend the augmented prompt.

3.3. Choice of PLMs

We choose three prominent PLMs to demonstrate our RAMP method and use their checkpoints from the Hugging Face hub for our experiments.
T5: We utilize T5 (Raffel et al., 2020) due to the fact that it is the standard model for D2T in most previous work (Kale and Rastogi, 2020).
Flan-T5: We use Flan-T5 (Chung et al., 2022), an instruction fine-tuned variant of the T5 model pre-trained on many tasks. We anticipate it to align better with our prompted input given that it was fine-tuned with instructions.
LED: We further include LED (Longformer-Encoder-Decoder; Beltagy et al., 2020), a transformer-based encoder-decoder PLM based on Longformer, which performs strongly on long documents.

For all our experiments, the maximal length of our input prompt is 3744 tokens, which is way longer than the context window size of T5 and Flan-T5 (i.e., 512 tokens). The primary reason for adding the LED model to our experiment is its ability to handle very large context lengths (i.e., 16K tokens).

Models	**BLEU**	**ROUGE**	**METEOR**	**PARENT**
Prompting				
Vicuna-13B 0-shot	1.34	14.58	21.99	8.72
Vicuna-13B 1-shot	17.88	23.98	34.87	11.51
Ours				
T5$_{RAMP}$	78.92	89.82	90.30	67.45
Flan-T5$_{RAMP}$	85.12	92.37	92.48	70.99
LED$_{RAMP}$	**91.76**	**96.07**	**94.92**	**74.92**

Table 2: Performance comparison of various models on the test split of the Drone dataset.

We also experimented with Vicuna-1.3-7B (Chiang et al., 2023), LLaMA-2-7B (Touvron et al., 2023) and Mistral 7B (Jiang et al., 2023), but we could not achieve satisfactory results using the same prompt settings. We speculate this was due to the prompt formatting issues discussed by Sclar et al. (2023). Or it could be that larger language models have stronger inductive biases which can not be easily altered with prompt-tuning on a limited amount of data (Chen et al., 2022a). We will further explore these models in our future work.

For zero-shot and one-shot baseline models, we use Vicuna (Chiang et al., 2023), an approximation to the current state-of-the-art PLM ChatGPT, which is a fine-tuned LLaMA model (Touvron et al., 2023) trained on 70K user-shared ChatGPT conversations. We aim to assess its performance in our specific context of drone data-to-text generation.

4. Result and Analysis

In this section, we present our experimental results for both automatic and human evaluation. Our goal is to provide a fair comparison study of different prompting methods along with RAMP. We also lay out a study comparing different models to test the consistency of RAMP method.

4.1. Automatic Evaluation

We use classical NLG automatic metrics including BLEU-4 (BLEU; Papineni et al., 2002), ROUGE-L F1 (ROUGE; Lin, 2004), METEOR (Banerjee and Lavie, 2005) for overall quality. In addition, we also use the PARENT (Dhingra et al., 2019) metric to identify hallucinations by evaluating the generated text against both references and the input data.

We first compare our RAMP method with prompting as the baseline method for low-resource D2T. For the prompting method, we employ 0-shot and 1-shot prompting approaches with one of the state-of-the-art PLMs, Vicuna-13B (Chiang et al., 2023). We follow Occam's razor for prompt engineering and use a fixed example for all input.[1] Table 2

[1] We cannot experiment with more examples because

Methods	BLEU	ROUGE	METEOR	PARENT
Flan-T5				
no example	47.28 ± 0.09	63.88 ± 0.09	63.01 ± 0.13	29.47 ± 0.04
fixed	62.66 ± 0.08	71.26 ± 0.02	71.56 ± 0.07	34.59 ± 0.01
retrieved	71.22 ± 0.10	81.13 ± 0.04	79.36 ± 0.10	46.27 ± 0.02
RAMP	**85.12** ± 0.05	**92.40** ± 0.03	**92.48** ± 0.11	**70.99** ± 0.02
LED				
no example	31.09 ± 0.28	54.43 ± 0.21	51.30 ± 0.26	36.46 ± 0.03
fixed	63.27 ± 0.16	77.80 ± 0.32	76.05 ± 0.11	47.60 ± 0.03
retrieved	72.84 ± 0.05	83.45 ± 0.20	80.39 ± 0.05	†69.09 ± 0.00
RAMP	**93.47** ± 0.02	**96.66** ± 0.04	**95.43** ± 0.03	**74.92** ± 0.02

Table 3: Effectiveness study of different prompting strategies, with a specific focus on the best performing models, Flan-T5 and LED. †The standard deviation is 0.003.

shows that our RAMP method outperforms both 0-shot and 1-shot prompting by a large margin significantly, Welch's t-test on three repeating experiments, $p < 0.01$. We see that although Vicuna models are bigger and better performing on various benchmarks, they perform poorly on our task. We notice that since Vicuna is fine-tuned as a chat assistant, it tends to generate longer outputs. Vicuna also achieves very low PARENT scores indicating severe hallucinations in its output. In addition, we also find that Flan-T5 outperforms T5 on all metrics significantly, Welch's t-test on three repeating experiments, $p < 0.01$.

We then compare different prompting strategies on the two best models Flan-T5 and LED from Table 2 using automated evaluation metrics in Table 3. Evidently, the more adaptive prompting strategies that fit the task outperform the more generic examples and continuous prompts, i.e., *no example* $<$ *fixed* $<$ *retrieved* $<$ *modular*. These results affirm the importance of a well-aligned prompt design that matches with data semantics.

Additionally, we also study the performance of various models on our task. Table 2 provides a comprehensive overview of the model performances. Among all the models, we observe that the LED models consistently achieve the highest scores across all evaluation metrics. We speculate the reason for these high scores is due to its ability to handle larger context inputs.

4.2. Human Evaluation

In this subsection, we provide the results of our human evaluation study. For this study, our aim is to better understand the generated handover messages beyond automatic metrics.

We conduct a human evaluation study on the test set to further understand the correctness and hallucination issues for all outputs. We asked two annotators to evaluate 166 generated texts each from six models. So we annotate 996 texts in total and each text has two labels. We also present Cohen's kappa scores κ for the inter-annotator agreement of each evaluation metric. The annotators evaluate these texts by applying judgments based on four binary properties.

Intrinsic hallucination: the generated text directly contradict the input data (Zhou et al., 2021);

Extrinsic hallucination: some content in the generated text is not grounded in the input data (Zhou et al., 2021);

Coverage: whether all the attributes in the input data are mentioned in the generated text (Jolly et al., 2022).

Correctness: whether the generated text is free from grammatical and factual errors (Howcroft et al., 2020). This metric overlaps with two hallucination aspects because we want to give an overall quality estimation.;

The results of this study are showcased in Table 4. We compare modular continuous prompts with the fixed 1-shot prompt and retrieved augmented prompt methods. We can see that modular continuous prompts surpass the performance of both the retrieved and fixed prompt settings. This observation lends further support to the effectiveness of the modular prompt approach, which is trained with distinct attribute-specific knowledge. Our findings are supported by the higher inter-annotator agreement scores (i.e., κ).

We notice that retrieval-augmented methods obtain lower Coverage, which could be the case that these models generate shorter texts because of lower Extrinsic Hallucination. So we also inspect the output length and find that retrieval-augmented methods indeed generate shorter texts and lead to lower coverage. In the meantime, the Intrinsic Hallucination is not correlated to the output length.

The modular prompt design has demonstrated a remarkable ability to elevate the performance of pre-trained language models (PLMs) in generating text outputs that are notably more accurate

the lengths of some input already exceed the context window size of Vicuna. Details of hyperparameters and prompt engineering are in the Appendix A.

	Intrinsic	Extrinsic			
Methods	**H** $\downarrow$	**H** $\downarrow$	**Correctness**	**Coverage**	**# Token**
Flan-T5					
fixed	27.11	5.72	64.16	**87.02**	112.23
retrieved	37.65	2.41	61.15	86.15	108.36
RAMP	17.77	2.41	75.30	81.03	105.46
LED					
fixed	37.65	7.83	59.64	82.23	107.60
retrieved	23.87	**1.52**	75.23	81.10	105.04
RAMP	**11.15**	2.41	**81.03**	81.33	105.90
κ	77.39	71.17	80.55	65.04	

Table 4: Human evaluation results. All numbers are percentages of samples in the test set that exhibit the corresponding property. ↓ means the lower the better. We also report the average numbers of tokens in the last column.

and contextually relevant. Significantly, the striking resemblance between the automatic evaluation metric outcomes in Table 3 and the human evaluation results in Table 4 serves to further underscore the robustness of our method and strengthens the credibility of our findings.

4.3. Self Assessment

To further understand the error patterns of our system, we analyzed the errors made by the system manually. We also wanted to understand why intrinsic hallucinations increased for retrieved prompts in Flan-T5 models and why we see coverage decreases for modular prompts.

Upon looking closely, we observed that whenever there are longer entities such as "*wall inside building*", "*bell-shaped statue*" mentioned in the input, the Flan-T5 model fails to copy them correctly in the case of retrieved prompts. For the coverage issue, although the numbers are lower for the Longformer models, it consistently only misses to mention "*PilotExperienced*" parameter. We believe these issues could be fixed by changing the finetuning hyperparameters.

In general, we found the following recurring patterns in the errors made by the models.

Repeated Entities and Tokens We observe that models tend to repeat some entities such as "drone droness" instead of "drone's", "bell in in in the drone's flight" instead of "bell in the drones' flight" while generating messages.

Partial Phrases Similarly, we also notice that the models sometimes fail to the full phrases. They rather copy partial phrases from the input such as "plant on Para" instead of "planted parapet" while generating messages.

We believe both of these issues could be fixed by adding the repetition penalties and using better token sampling methods during inference.

5. Conclusion

In this work, we present a retrieval-augmented modular prompt design – RAMP, for a low-resource D2T generation task. We utilize the drone sensor dataset that is small and diverse in terms of data records. Both the automated evaluation and the human evaluation results demonstrate the effectiveness of augmented prompts, especially the modular augmented prompts. The trainable design of RAMP serves as a vital link for adapting specific input data formats and augmented examples, facilitating the seamless addition of domain-specific knowledge into the PLMs generation abilities. RAMP also shows crucial improvements in hallucination. As a result, our RAMP method presents a promising solution to enhance the versatility and robustness of D2T systems in real-world applications.

6. Acknowledgement

This research was funded by DFG grant 389792660 as part of TRR 248 – CPEC.[2] We sincerely thank the anonymous reviewers for their insightful comments that helped us to improve this paper. We thank Nina Shvetsova for her help with the Vicuna models. We also thank our student assistant Khushboo Mehra for evaluating the model output.

7. Ethical Considerations

The human evaluation study presented in this work is carried out by two student assistants at the university. They were paid fairly as per the university payment standards. We also advise evaluating our methods on the appropriate validation sets before using them for other domains and datasets.

[2] https://perspicuous-computing.science

8. Limitations

Our paper is a small and concentrated contribution targeting hallucinations in a low-resource D2T task. We only conducted experiments on one data set. The scale of this dataset is relatively small compared to other standard D2T datasets. Investigating the effect of our RAMP on other D2T datasets is a topic for future investigation.

9. Bibliographical References

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

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

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 *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA.

Shuyang Cao and Lu Wang. 2022. Time-aware prompting for text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 7231–7246, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ernie Chang, Alisa Kovtunova, Stefan Borgwardt, Vera Demberg, Kathryn Chapman, and Hui-Syuan Yeh. 2022. Logic-guided message generation from raw real-time sensor data. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6899–6908, Marseille, France. European Language Resources Association.

Guanzheng Chen, Fangyu Liu, Zaiqiao Meng, and Shangsong Liang. 2022a. Revisiting parameter-efficient tuning: Are we really there yet? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2612–2626, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Hailin Chen, Amrita Saha, Shafiq Joty, and Steven C.H. Hoi. 2022b. Learning label modular prompts for text classification in the wild. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1677–1690, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *Large Model Systems Organization*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Bhuwan Dhingra, Manaal Faruqui, Ankur Parikh, Ming-Wei Chang, Dipanjan Das, and William Cohen. 2019. Handling divergent reference texts when evaluating table-to-text generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4884–4895, Florence, Italy. Association for Computational Linguistics.

David M. Howcroft, Anya Belz, Miruna-Adriana Clinciu, Dimitra Gkatzia, Sadid A. Hasan, Saad Mahamood, Simon Mille, Emiel van Miltenburg, Sashank Santhanam, and Verena Rieser. 2020. Twenty years of confusion in human evaluation: NLG needs evaluation sheets and standardised definitions. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 169–182, Dublin, Ireland. Association for Computational Linguistics.

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

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Shailza Jolly, Zi Xuan Zhang, Andreas Dengel, and Lili Mou. 2022. Search and learn: improving semantic coverage for data-to-text generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10858–10866.

Mihir Kale and Abhinav Rastogi. 2020. Text-to-text pre-training for data-to-text tasks. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 97–102, Dublin, Ireland. Association for Computational Linguistics.

Moniba Keymanesh, Adrian Benton, and Mark Dredze. 2022. What makes data-to-text generation hard for pretrained language models? In *Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 539–554, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Yanchen Liu, Timo Schick, and Hinrich Schtze. 2023. Semantic-oriented unlabeled priming for large-scale language models. In *Proceedings of The Fourth Workshop on Simple and Efficient Natural Language Processing (SustaiNLP)*, pages 32–38, Toronto, Canada (Hybrid). Association for Computational Linguistics.

Yutao Luo, Menghua Lu, Gongshen Liu, and Shilin Wang. 2022. Few-shot table-to-text generation with prefix-controlled generator. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6493–6504, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Nick McKenna, Tianyi Li, Liang Cheng, Mohammad Javad Hosseini, Mark Johnson, and Mark Steedman. 2023. Sources of hallucination by large language models on inference tasks. *arXiv preprint arXiv:2305.14552*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

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

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

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

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2023. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. *arXiv preprint arXiv:2310.11324*.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation.

In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3784–3803, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *ArXiv*, abs/2302.13971.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

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

A. Appendices

A.1. Prompt Engineering

In this section, we discuss the prompts that we designed. We experiment with prompts with role or without role following previous work (Ouyang et al., 2022). Here are two examples.

Without role. *Here is the raw sensor data from a drone: INPUT_DATA. Write the handover messages that only include crucial situations for this data.*

With role. *You are an assistant that generates handover messages for a semi-autonomous drone used to communicate with humans during the drone flight. Now I give you the raw sensor data and you will need to generate handover messages that only include crucial situations. Here is the input raw sensor data from a drone: INPUT_DATA. Write the handover messages that only include crucial situations for this data.* Experiments show that adding roles in our prompts doesn't improve performance.

A.2. Activated Logic Expressions

The retrieval augmentation method 3.1 uses the logic expression to retrieve examples that are similar to the input sensor data. This expression represents the application of ontology axioms to determine critical situations in the input data. In a single data record, numerous critical situations might emerge, each associated with various applicable expressions. The DL expressions act as filters, segmenting and categorizing the input data according to these axioms. An example of how these expressions are applied to the input is provided in Table 6. The complete set of description logic expressions and their most representative examples are in Table 7.

A.3. Number of Continuous Prompt Tokens

We were interested in the impact of the length of the continuous prompt on performance. Table 5 shows the effect of varying the length of the continuous prompt lengths between 5 and 60 tokens for the Flan-T5 and the LED models. The experiments use the model setting where the continuous prompt is augmented by the retrieved few-shot examples.

Our experimental results, which include updated findings, indicate a positive correlation between the number of continuous prompt tokens and the performance of our model. Specifically, we observed that the LED models exhibit a higher PARENT score as the number of continuous prompt tokens increases. Notably, this increase exhibited a slowdown after the number of continuous prompt tokens exceeded 20. On the other hand, for the Flan-T5 model, we observed a dramatic increase in the PARENT score when the number of continuous prompt tokens exceeded 40. These observations demonstrate the intricate relationship between continuous prompt length and model performance, shedding light on optimal configurations for different model variants. By demonstrating the beneficial alignment between the pretrained large language model and our data-to-text dataset, the findings suggest that even with 60 continuous tokens, the model can effectively adapt to the sampling space of the drone dataset.

A.4. Experiment Settings

We utilize checkpoints from Hugging Face for the following models: `T5-base`[3], `flan-t5-xl`[4], `led-base-16384`[5], and `vicuna-13b-v1.3`[6]. These experiments are performed on a GPU server equipped with 8 Tesla A100 cards, training each model using three randomly selected seeds. We

[3] https://huggingface.co/t5-base

[4] https://huggingface.co/google/flan-t5-xl

[5] https://huggingface.co/allenai/led-base-16384

[6] https://huggingface.co/lmsys/vicuna-13b-v1.3

# tokens	BLEU	ROUGE	METEOR	PARENT
Flan-T5				
5	67.19	79.61	78.00	44.07
10	70.04	80.51	78.75	45.77
20	71.22	81.13	79.36	46.27
30	70.75	81.26	79.14	52.17
40	71.06	81.27	79.15	54.57
50	71.22	81.13	79.36	71.76
60	**72.33**	**81.87**	**79.47**	**72.27**
LED				
5	67.19	79.61	78.00	51.38
10	70.04	80.51	78.75	63.18
20	68.14	81.89	78.62	69.09
30	72.34	83.28	80.39	71.15
40	71.23	83.28	80.24	70.51
50	72.05	83.60	80.41	70.78
60	**75.61**	**84.92**	**81.76**	**71.16**

Table 5: Impact of varying continuous prompt lengths on Flan-T5 and LED models with retrieved few-shot examples.

Name	Notation	Example
DL-filtered data	x	{ "time_stamp": "0:05", "name": "castle", "Distance": 2.5 }
Activated expression	E	[Distance : 2.5 $\sqsubseteq$ [Distance $\leq$ 3.0m] $\sqsubseteq$ **VeryClose**] [Time_stamp : "0:05" $\wedge$ Name : "castle" $\wedge$ **VeryClose**] $\sqsubseteq$ [VeryClose.Object] [$\exists$ VeryClose.Object] $\sqsubseteq$ RiskOfPhysicalDamage
Retrieved input	x_r	{ "time_stamp": "0:01", "name": "gravestone 1", "Distance": 3.0 }
Retrieved output	y_r	(0:01) The drone is facing the risk of physical damage. There is a gravestone in the drone's flight path at a distance of 3.0m.
Augmented input	x'	[Continuous Prompts], x_r, y_r, x

Table 6: Example of logic-guided retrieval. The preprocessed input x aligns with the logic expression E, representing a scenario of a drone's close proximity to an object. An input x_r resembling the logic expression E is retrieved from the training data, along with its corresponding text y_r. These elements are amalgamated to form a retrieved example, which in turn enhances the continuous prompt through augmentation.

run each model using three different random seeds including 3407, 42, and 1223. The evaluation metrics, including both automatic and human evaluation, average the outcomes of three experimental runs under the same prompt configurations (fixed, retrieved, modular) to ensure reliability.

Description Logic	Activated Expression	Retrieved Example
broken frame	[Altitude $\geq$ **Flying**] $\wedge$ [Normal_frame $\sqsubseteq$ **Normal**]	"The drone is flying with a damaged frame."
nearby moving object	[Distance $\leq$ **Near**] $\wedge$ [Object_Type $\sqsubseteq$ **Moving_objects**]	"(0:01) The drone is risking physical damage. It's flying too close to the moving car at a distance of 3.0m."
reachable object inpath	[Distance $\leq$ **Reachable**] $\wedge$ [Inpath **True**]	"(0:05) The drone is facing the risk of physical damage. There is a car in the drone's flight path at a distance of 0.3m."
empty battery	[Altitude $\geq$ **Flying**] $\wedge$ [Battery_level $\leq$ **empty_battery**]	"The flying drone is runing out of battery with only 20% charge."
low battery & strong wind	[WindSpeed $\geq$ **Strong_wind**] $\wedge$ [Battery_level $\leq$ **empty_battery**]	"The drone is flying in a strong wind of 18m/s with a low battery level at 30%."
inexperienced nearby object	[Distance $\leq$ **Near**] $\wedge$ [PilotExperienced **False**]	"(0:00) The drone is facing the risk of physical damage. It's flying too close to Bird1 at a distance of 0.5m, and the pilot is not experienced."
low battery & low temperature	[Temperature $\leq$ **Low_temperature**] $\wedge$ [Battery_level $\leq$ **Low_battery**]	"The drone is flying in low temperature at 0 degree with a low battery level at 40%."
low battery & high altitude	[Altitude $\geq$ **High_Atitude**] $\wedge$ [Battery_level $\leq$ **Low_battery**]	"The drone is flying so high at 80m height with a low battery level at 40%."
precipitation	[waterproof **False**] $\wedge$ [Weather $\sqsubseteq$ **Precipitation**]	"The drone is facing a risk of internal damage as it's flying in gloomy weather and not waterproof."
gloomy & high altitude	[Altitude $\geq$ **High_Atitude**] $\wedge$ [Weather $\sqsubseteq$ **Gloomy**]	"The drone would be damaged physically as it's flying at a high altitude of 90m in a gloomy weather.
out of range while low-battery	[Distance_from_control $\geq$ **Almost_out_range**] $\wedge$ [Battery_level $\leq$ **Low_battery**]	"The drone has only 20% battery and is 4490m away from the remote control."
water surface & low altitude	[Altitude $\leq$ **Low_altitude**] $\wedge$ [flying_over $\sqsubseteq$ **Water**]	"The drone is facing a risk of physical damage as it's flying over water surface at a very low altitude of 0m height."
low altitude & fast speed	[Altitude $\leq$ **Low_altitude**] $\wedge$ [DroneSpeed $\geq$ **Fast_speed**]	"The drone is facing the risk of physical damage. It's flying in a high speed of 16m/s and low altitude of 0m."
low visibility & nearby object	[Low_visibility **True**] $\wedge$ [Distance $\leq$ **Near**]	"(0:00) The drone might get physical damage. It's flying with a low visibility, and too close to tube at a distance of 0.5m."
very close to human	[Object_Type **Human**] $\wedge$ [Distance $\leq$ **Very_close**]	"(0:00) The drone is flying very close to a human at a distance of 0.5m, and might cause human injury."
upsidedown & inexperienced	[upside_down **True**] $\wedge$ [PilotExperienced **False**]	"The drone is risking physical damage for it's flying upsidedown and the pilot is not experienced."
indoor & nearby human	[Indoor **True**] $\wedge$ [Distance $\leq$ **Near**] $\wedge$ [Object_Type **Human**]	"The drone might cause human damage, it's flying indoor, and there is a person only 3.0m away."

Table 7: Descriptive logic and retrieved examples.