Exploring Abductive Reasoning in Language Models for Data-to-Text Generation

1st Kristýna Onderková
School of Computer Science
University of Galway
Galway, Ireland
navitas.onderkova@gmail.com

2nd Matthias Nickles
School of Computer Science
University of Galway
Galway, Ireland
matthias.nickles@universityofgalway.ie

***Abstract*—Abductive reasoning remains underexplored in language models despite its everyday human use, effectiveness in handling incomplete information, and use in automated planning. We present a data-to-text generation pipeline that prompts language models with abductive tasks to investigate its applicability. We show its utility in content selection, though generating a discourse plan for selected content presents challenges for non-fine-tuned language models. The three-stage pipeline allows for the deployment of more suitable models for different stages (reasoning and realization). This work highlights the potential of symbolic reasoning approaches in enhancing language models.**

***Index Terms*—Language Models, Abductive Reasoning, Natural Language Generation, Data-to-Text Generation, Planning**

I. Introduction

Advancements in language models (LMs), in particular Large Language Models (LLMs), have driven their widespread adoption. However, limitations such as the lack of robust reasoning and planning abilities [1], coupled with issues like hallucinations and omissions [2], restrict their applicability in data-to-text (D2T) tasks across rigorous domains like meteorology, medicine, law, and journalism [3]. Classical Natural Language Generation (NLG) approaches [4] avoid accuracy issues with template and rule-based methods within a pipeline architecture but are labour-intensive. Therefore, [5] segmented neural models into separate pipeline stages and introduced a symbolic text planning stage before LM-based text generation, resulting in more faithful and controllable results.

LLMs exhibit remarkable properties, including in-context learning and multistep reasoning [6], without requiring additional training or parameter updates [1]. Prompting can invoke these abilities, for example, through engineered instructions (zero-shot) and task-specific examples (few-shot). Prompts, typically derived from templates with input placeholders as *"Czech: [x] English: "*, prime the model to generate more accurate predictions based on the template and input probabilities [6]. However, their performance tends to lag behind specially trained state-of-the-art models, leading to the development of various prompting techniques and strategies [6].

Prompting can enhance LMs' reasoning abilities by employing symbolic semantic representation methods [6], [7]. Logical reasoning includes deduction, induction, and abduction (abductive reasoning) [8]. The latter two are forms of defeasible inference, meaning their conclusions may not always be deductively valid [8]. However, this enables them to handle incomplete observations and generate new ideas [9]. Deduction has been extensively studied [10], but research on abduction in NLG remains limited, e.g., [7], [10]. A formal definition of abduction in [11] requires abductive hypotheses to be consistent with the logical theory representing a given domain and logically entail all observations. As an example of abduction, when finding a broken mug, we can abductively infer that it was not broken by a unicorn (as they do not exist) but was probably broken by our cat, given the rarity of earthquakes in our region. Abductive reasoning can be used for planning, where the goal is the assumed observation (a baked cake) and the plan steps (baking, mixing, gathering ingredients) are inferred as an "explanation" for the goal.

This work examines the use of abductive reasoning in D2T generation, as abduction seeks the most plausible explanations based on given data and rules. To explore this approach, we establish a D2T pipeline, focusing on the initial stage of the classical NLG approach [4], where we can apply reasoning. We explore the application of abduction to select a set of messages in a content determination task. This approach facilitates the addition of new rules, as abduction can integrate novel information and refine conclusions [9]. Then, we investigate how we can structure these selected messages in the discourse planning task. We combine the latter two stages of the classical NLG pipeline [4] into a single language realization stage, leveraging the suitability of LMs for this task.

II. Related Work

Prior research on multi-stage LM text generation includes [5], [12], and fine-tuned models from [13], [14]. [15] focuses on discourse planning, and [16] develops a zero-shot prompted pipeline. To the best of our knowledge, no prior work has directly applied abductive reasoning in LMs for text generation.

There are several datasets for abductive reasoning in LMs, including *AbductionRules* [17], *D*-Ab* [18], and *ART* dataset [9], which inspired our prompt formulation. The *ART* dataset includes two subsequent observations (past and future), requiring the LM to select or generate the most plausible hypothesis

979-8-3503-6021-9/23/$31.00 ©2023 IEEE

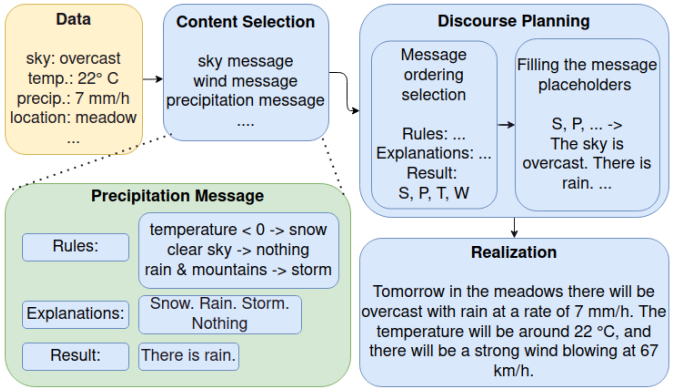

Fig. 1. Schema of the implemented D2T pipeline with an example of an abductive reasoning task for content selection

connecting the observations. Their model achieved 68.9% accuracy in selecting and 45% in generating hypotheses, notably below human performance at 91.4% and 96%, highlighting the reasoning challenges. Two related works fine-tune LMs for this dataset: [19] uses commonsense temporal reasoning, and [20] introduces a *multi-level knowledge-aware reasoning* approach, leveraging knowledge graphs at the event-level and word-level. Researchers also explored abduction in LMs for backward reasoning, question answering, generating plausible proofs, and assessing the plausibility of hypotheses [10].

III. Methodology

Our pipeline (Fig. 1) consists of three stages: content selection, discourse planning, and realization. Inspired by [9], we employ abductive prompts (Table I) that contain crafted logical rules representing the domain, observations for each variable, and possible explanations. The model selects the best explanation as the chosen message. In discourse planning, we apply a similar approach to pick the optimal message sequence (indicated by letters) and replace these markers with actual messages. That flows into the realization stage, which we explore only briefly as it diverts from our abduction focus.

While experimenting with several models, such as *LLaMA* (3B), *GPT-2 XL*, and *Falcon Instruct 7B*, only *Flan-T5* demonstrated usable reasoning capabilities. The pipeline, built within the LangChain framework [21], prompts the *Flan-T5 XXL* model and *Falcon Instruct 7B* (for realization) through the HuggingFace's inference API [22] with a conservative temperature setting of 0.1 and a maximum token limit of 64. The code is available on GitHub [23].

For testing the pipeline, we chose the weather domain due to its popularity and our familiarity with it. To concentrate on exploring the abductive approach without getting caught up in complex implementations required for existing datasets, we created a simplified weather dataset. It comprises six aspects (columns) and 600 instances (rows) randomly generated for our study. The aspects with their possible values are sky state (clear, cloudy, overcast), temperature (-10 to 30°C), precipitation (0 to 20 mm/h), wind (0-100 km/h), animals (cat, bird, fish) with varying weather preferences, and locations (beach, city, meadow, mountain) for potential weather predispositions.

TABLE I
Illustration of prompts used in the pipeline

Task	
structure of content selection prompts	The reasoning task is to select the more probable explanation, given the rules. There are x variables: [VAR:a], [VAR:b], ... **Rules**: ... The **explanations** are: ... [VAR:a] is a. [VAR:b] is b. ... Answer:
message order	The reasoning task is to select the best order of events... (rest same as content selection)
substitution	Replace letters in a given sequence with their full corresponding sentences...
realization	Generate a fluent weather forecast in future tense from provided informations...
Rules	
sky	place=beach & animal=cat $\Rightarrow$ mist. place=city or animal=bird $\Rightarrow$ smog.
wind	Wind $< 18 \Rightarrow$ no wind. Wind $> 50 \Rightarrow$ strong wind.
precipitation	temperature $< 0 \Rightarrow$ snow. Precipitation=0 $\Rightarrow$ no precipitation. Sky=clear $\Rightarrow$ no precipitation. Rain & place=mountains $\Rightarrow$ storm.
order	Animal=cat $\Rightarrow$ T, S, P, W. Sky=clear $\Rightarrow$ P, S, T, W. Animal=bird $\Rightarrow$ W, S, P, T. Usually S, P, T, W.
Explanations	
sky	Mist. *VAR:sky*. Smog.
precipitation	No precipitation. Snow. Rain. Storm.
wind	Wind. No wind. Strong wind.
ordering	S, P, T, W; W, S, P, T; P, S, T, W; T, S, P, W

The full prompts' formulations can be found on Github [23]

A. Content Selection

Given the limited capabilities of the non-fine-tuned model, it seems desirable to divide the process into smaller steps, assigning at least one prompt per atomic message. Therefore, we crafted three core messages (Table I): sky coverage, precipitation, and wind conditions. By removing different parts of the *sky* prompt, we assessed elementary logic usage (equality, implication, and, or). The *wind* prompt tested basic number comparisons and introduced an abductive reasoning step, as wind occurrence is not explicitly stated. The most intricate *precipitation* prompt required two reasoning steps, involving abducted knowledge for abducting the solution. We explored several prompt versions based on a simplified precipitation prompt without an extra abductive step. We modified it into a reframed generative version without explanation options, a version excluding variables that enabled the use of models' world knowledge, and a prompt with additional explanations. These explanations assessed whether the model exhibits confusion when presented with irrelevant solutions (*"pizza"*) and more plausible ones (*"shower"* and *"drizzle"*).

We implemented all prompts in a zero-shot setting, as more complex methods (few-shot, Chain-of-Thought) did not enhance results. Explicitly indicating variables (such as temperature and precipitation) with formulation *[VAR:variable]* proved beneficial. We also attempted to develop a prompt to remove the wind message when the wind was minimal, as it is unnecessary for reporting. However, this posed challenges,

causing either excessive or insufficient content removal. Consequently, we excluded it from the pipeline.

B. Discourse Planning

For arranging messages according to specific preferences, it would be desirable to generate such a sequence. However, the *Flan-T5* model proved inadequate for this purpose. We attempted to implement a genuine planning approach through simple rules that specified which sentence to place first, last, or before/after another. Yet, the only effective strategy was to position a single sentence at the beginning. Additionally, the relocated sentence often remained in its original position.

Eventually, we employed a content selection stage approach, simplifying messages to symbols (e.g., "S" for sky). The model inferred a symbol sequence by abductive reasoning as an explanation according to rules (Table I). We explored a case where two orderings were possible. Then, we replaced the chosen orderings with the corresponding messages from the content selection stage. Finally, we generated forecasts using both *Flan-T5* and *Falcon Instruct* models.

IV. Results and Interpretation

Our D2T pipeline prototype with abductive inference prompts for a simplified use case serves to showcase feasibility and investigate its capabilities and limitations. Table II presents the accuracy achieved in our experiments.

A. Content Selection

Content selection demonstrates feasibility despite lower accuracy. *Sky* prompts show bias toward better weather conditions, evident in the prompt version without logical constraints, where *clear sky* explanations are chosen in about 1/3 of instances. The prompt's complexity is comparable to the *simple precipitation* prompt, which is roughly 6% more accurate. Introducing the *and* constraint reduces bias against *mist* to 1/4 and adds a 1/5 bias toward *cloudy* for *smog* explanations. The *or* constraint minimally impacts bias, and both constraints together appear to behave as a simple combination. Simulated weather conditions yield non-uniform outcomes, posing challenges for quantitative evaluation favouring a more uniform dataset. Yet, this underscores the intricacy of reasoning about equality and elementary logical implications with current LMs.

Contrary to our expectations, when the *and* condition is applied, *mist* is chosen if it meets only one of its rules, functioning like an *or* operation. We are uncertain whether the model genuinely performs the *or* operation, which prompt refinement could correct, or if it just finds the variable and explanation in one sentence of rules. The *or* constraint seems ineffective. In instances with both *bird* and *beach* variables, the model may select two different explanations. In such cases, it consistently favours the earlier rule, suggesting that the rule sequence influences how the model treats them.

The *wind* prompt adeptly abducted information not explicitly given and showcased the model's usable arithmetic skills in number comparisons. Occasional errors (1.5%) emerged only when the wind speed was precisely 50 km/h, suggesting potential improvement with a more precise prompt. In complex scenarios, a calculator tool invoked by the LM framework would enhance reliability.

TABLE II
Evaluation of Reasoning Abilities with Proposed Prompts

sky prompts		*and*	*or*	*none*	**wind**
acc. [%]	50.5	64.0	63.2	81.5	98.5

precipitation		*simple*	*knowledge*	*generative*
acc. [%]	93.7	87.7	65.3	83.5

precipitation	*pizza*	*drizzle*	*shower*	**ordering**	**substitution**
acc. [%]	64.8	66.3	65.5	100	77.3

The *simple precipitation* prompt exhibits minor numerical inaccuracies, favouring snow at temperatures just above freezing, leading to a 2% decrease in accuracy, affecting 6% of rain cases. A more substantial issue is the infrequent application of the last rule (clear sky), causing an 11% accuracy drop and affecting about 18% of cases where it erroneously predicts precipitation. This observation highlights a key challenge: the model's treatment of rules varies based on their order.

The *full precipitation* prompt improves accuracy by 9% due to improved use of the clear sky rule. Erroneous precipitation selection now causes only a 2% accuracy drop. Numerical inaccuracies in comparing temperatures when selecting snow over rain improve slightly, affecting 3% of rain cases. The storm rule with the *and* condition performs well, rarely selecting a storm without precipitation. However, the model favours storm over snow in 22% of cases, resulting in a 2% accuracy decline. This suggests an ability to abduct from previously inferred information, although it only considers precipitation presence without distinguishing between rain and snow.

The model's reliance on its *knowledge*, in prompts without explicit variable declaration, leads to a 22% increase in errors, notably neglecting the last rule on clear skies in 68% of its cases, resulting in a 19% accuracy drop. It also mistakenly predicts precipitation in 19% of its cases and snow in 22%, with temperatures up to 9°C, compromising numerical accuracy.

The *generative* prompt performs well, especially in identifying *no precipitation* from the rule about clear skies, with a 4% false negative rate for snow causing a 3% accuracy drop. However, it misclassifies rain as snow in 46% of rain cases, leading to a 14% accuracy decline. The discrepancy is not due to arithmetic errors, as evidenced by its selection of snow with high temperatures up to 29°C. While the model is adept at recognizing all snow cases, it struggles to abduce rain independently without a direct mention in the prompt.

While the model is proficient in choosing the more likely option (selected only *shower* in 3% of cases) when presented with an *additional explanation*, all those explanations decreased the accuracy by about 22% compared to the *simple* prompt. It chose the *snow* option instead of *rain* in about 14% of cases, reducing accuracy by 4%. Moreover, in half of the instances without precipitation, the model incorrectly predicts it, diminishing accuracy by 30%. Hence, it is advisable to avoid introducing additional hypotheses.

B. Discourse Planning

As mentioned, we struggled to create a proper planning prompt and instead adopted the content selection approach. The *ordering* prompt consistently yielded accurate results, irrespective of providing all possible orderings in the rules (*usually S, P, T, W.*). This highly symbolic prompt avoided introducing learned biases, showcasing the *Flan-T5* model's capability for simple abductive reasoning.

However, translating the symbolic plan into actual messages using our *substitution* prompt proved suboptimal. It failed to convert letters into sentences in approximately 12% of cases, primarily affecting the *PSTW* ordering in 80%. The result consistently included temperature and precipitation. Nevertheless, it omitted sky information (from PSTW ordering) in 7% of cases and wind details (from PSTW or SPTW) in 5% of cases, suggesting a potential link to a specific prompt formulation, offering a possible solution.

C. Realization

The *Flan-T5* model, while factually accurate, generates highly succinct responses. It encounters linguistic hurdles in tasks such as converting sentences to the future tense, choosing suitable prepositions, and crafting fluent forecasts. Leveraging the strengths of different models becomes valuable in this context, with the *Falcon* model demonstrating proficiency in verbal tasks and seamlessly merging messages into sentences.

The *Falcon*, achieving complete accuracy in about half of the cases, exhibits four common errors with comparable frequency. One involves substituting common phrases like *a gentle breeze* for *wind*, stemming from our simplified use case. This error is likely resolvable by selecting a suitable adjective during content selection, eliminating the need for hallucinations. Another error arises similarly from simplification, as forecasts usually place sky-related information at the beginning. Despite our custom ordering, the model occasionally follows this pattern, and fine-tuning could likely address this. Addressing the last errors, involving reporting an incorrect sky value and omitting some information, could be likely tackled by gradually refining the messages through incremental reformulations and integrating them cohesively.

V. Conclusions and Future Work

We demonstrated the employment of abductive reasoning in a data-to-text generation pipeline and outlined its construction. This approach showed promise despite suboptimal accuracy metrics. Abductive reasoning is functional for content selection (choosing from possible solutions), and the model performs adequately on this task. While the model effectively identifies irrelevant solutions, their addition reduces the overall accuracy. The non-fine-tuned model sometimes struggles with our logic formulations and appears to treat rules differently based on their order. The model's internal knowledge can hinder accuracy, and it holds some biases affecting the outcomes, but we partially mitigated this by explicitly indicating variables. Discourse planning is challenging and better suited for generating solutions where the model is less capable. Moreover, employing one model for reasoning tasks and another for text realization is beneficial.

Several avenues for future research emerge from this study. To address our work's limitations, testing larger language models and fine-tuning them would be valuable. Building on our findings, exploring how the model's treatment of rules varies with their sequence and delving deeper into discourse planning strategies holds merit. On the application front, integrating time-specific messages and exploring the abduction approach for personalizing generation shows potential.

References

[1] S. Bubeck *et al.*, "Sparks of artificial general intelligence: Early experiments with gpt-4," *arXiv preprint arXiv:2303.12712*, 2023.

[2] O. Dušek, D. M. Howcroft, and V. Rieser, "Semantic noise matters for neural natural language generation," *arXiv preprint arXiv:1911.03905*, 2019.

[3] B. Peng *et al.*, "Check your facts and try again: Improving large language models with external knowledge and automated feedback," *arXiv preprint arXiv:2302.12813*, 2023.

[4] E. Reiter and R. Dale, "Building applied natural language generation systems," *Natural Language Engineering*, vol. 3, no. 1, pp. 57–87, 1997.

[5] R. Puduppully, L. Dong, and M. Lapata, "Data-to-text generation with content selection and planning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 6908–6915.

[6] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–35, 2023.

[7] S. Qiao *et al.*, "Reasoning with language model prompting: A survey," *arXiv preprint arXiv:2212.09597*, 2022.

[8] D. Walton, *Abductive reasoning*. University of Alabama Press, 2014.

[9] C. Bhagavatula *et al.*, "Abductive commonsense reasoning," *arXiv preprint arXiv:1908.05739*, 2019.

[10] F. Yu, H. Zhang, and B. Wang, "Nature language reasoning, a survey," *arXiv preprint arXiv:2303.14725*, 2023.

[11] T. Eiter and G. Gottlob, "The complexity of logic-based abduction," *Journal of the ACM (JACM)*, vol. 42, no. 1, pp. 3–42, 1995.

[12] A. Moryossef, Y. Goldberg, and I. Dagan, "Step-by-step: Separating planning from realization in neural data-to-text generation," *arXiv preprint arXiv:1904.03396*, 2019.

[13] A. Balakrishnan, J. Rao, K. Upasani, M. White, and R. Subba, "Constrained decoding for neural nlg from compositional representations in task-oriented dialogue," *arXiv preprint arXiv:1906.07220*, 2019.

[14] Y. Su, D. Vandyke, S. Wang, Y. Fang, and N. Collier, "Plan-then-generate: Controlled data-to-text generation via planning," *arXiv preprint arXiv:2108.13740*, 2021.

[15] T. C. Ferreira, C. van der Lee, E. Van Miltenburg, and E. Krahmer, "Neural data-to-text generation: A comparison between pipeline and end-to-end architectures," *arXiv preprint arXiv:1908.09022*, 2019.

[16] Z. Kasner and O. Dušek, "Neural pipeline for zero-shot data-to-text generation," *arXiv preprint arXiv:2203.16279*, 2022.

[17] N. Young, Q. Bao, J. Bensemann, and M. Witbrock, "Abductionrules: Training transformers to explain unexpected inputs," *arXiv preprint arXiv:2203.12186*, 2022.

[18] O. Tafjord, B. D. Mishra, and P. Clark, "Proofwriter: Generating implications, proofs, and abductive statements over natural language," *arXiv preprint arXiv:2012.13048*, 2020.

[19] R. Zandie, D. Shekhar, and M. Mahoor, "COGEN: Abductive commonsense language generation," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Jul. 2023, pp. 295–302.

[20] F. Mu and W. Li, "Enhancing text generation via multi-level knowledge aware reasoning." IJCAI, 2022.

[21] C. Harrison. LangChain. [Online]. Available: https://github.com/hwchase17/langchain

[22] I. Hugging Face. Hugging Face. [Online]. Available: https://huggingface.co/

[23] K. Onderková. MSc Diploma Thesis Code on Github. [Online]. Available: https://github.com/Kristyna-Navitas/PromptingLanguageModelsforAbductiveReasoningTasks