Towards Human Readable Explanations

Kha Luân Feltus
UCLouvain, Belgium

Victor Poucet
UCLouvain, Belgium

Hélène Verhaeghe
UCLouvain, Belgium

Abstract

LLMs and their capabilities to handle natural language are increasingly used as an intuitive interface between non-expert users and complex algorithms. This is facilitated by the recent introduction of the Model Context Protocol (MCP). In this work, the LLMs facilitate the use of explanation algorithms by translating their outputs back into natural language tailored to the user.

2012 ACM Subject Classification Author: Please fill in 1 or more \ccsdesc macro

Keywords and phrases Dummy keyword

1 Introduction

Large Language Models are increasingly being studied to automatically transform a natural-language description of a combinatorial optimization problem into a model [3, 7, 8, 9, 10], designing heuristics [2, 11, 12]. These techniques are primarily designed for non-experts to help them model and solve their problems.

These models can then be solved by a solver. However, it has been shown that non-expert users often do not trust overly complex automatic systems. This is why algorithms such as stepwise explanation [1] have been designed. Such an algorithm provides explanations alongside the propagation to explain the deductions made by the solver step by step.

However, the propagation can also lead to an unsatisfiable part of the search space. In addition, such models made by non-experts can sometimes be totally unsolvable for various reasons. First, the LLM used for modeling can have introduced an error. Secondly, as non-experts, they could have designed an overconstrained problem without knowing it. To identify the reasons why such a model or a subspace of the search space is unsat, one can use other types of explanation techniques [4], such as Minimum Unsatisfiable Subsets (MUS) [6].

Unfortunately, for non-expert users, these explanation techniques produce outputs in a mathematical-like form, which is often difficult to understand. The goal of this work (still in progress) is to investigate (i) how LLMs can translate these mathematical-like explanations into natural-language explanations, and (ii) how we can integrate explanability tools seamlessly within the chats of LLMs.

To achieve the goal, a simple platform is currently being designed. Its aim is to provide the LLM with access to explanation algorithms via Model Context Protocol (MCP) servers. Once a tool returns an explanation, the LLM automatically translates it for the user, dynamically adjusting to the user's level of understanding.

2 Integration of Explanability Tools within LLM Chatbots

Model Context Protocol (MCP) [5] helps standardize the interface between AI applications and other services, facilitating interactions between LLMs and other services. This standardization allows practitioners to easily build tools usable by the LLMs. It improves reliability on outputs to questions requiring actual reasoning by delegating them to such tools. In this context, the incorporation of explainability tools into LLM chatbots is studied by creating MCP servers that can invoke them.

3 Translation of Explanations to Natural Language

The translation of mathematical-like explanations into natural-language explanations that are understandable to non-expert users also depends strongly on the users themselves. As you might not explain something to a child the same way you would to an adult, the LLM's translation should adapt to the user. This is why various levels have been defined.

▶ **Example 1.** (Stepwise explanation on a 4 by 4 Sudoku) To illustrate the level, a very simple 4-by-4 Sudoku puzzle is used. On this sudoku, a line already contains the values 1, 2, and 3, the propagation leads to assigning the 4 to the remaining cell. This is the translation by the LLM of the stepwise explanation returned in some of the various levels defined:

- Level 1 - Children 6-8 years old: *In this row, the numbers 1, 2, and 3 are already there, there is no more room for 4!*
- Level 3 - Young teen 12-14 years old: *Given the constraint of row, each value can only appear once per row. Row 2 already has 1, 3, and 4, we can thus deduce that cell (2,2) has value 2.*
- Level 4 - Teen 15-17 years old: *The column constraint excludes values 1, 3, and 4 as already present. The bloc constraint also excludes 3. The unique remaining value in the domain of (2,2) is thus 2.*
- Level 6 - University level, beginner: *By propagation of the unicity constraints (row, column, bloc), the domain of variable (2,2) is reduced to a singleton* $\{2\}$*. The assignment Value(2,2) = 2 is a direct consequence of arc consistency.*

This *understanding level* parameter can be dynamically determined, either by inferring the education level of the user base from some prior questions. It can also be adjusted automatically downward whenever the user requests more precision or upward when the user requests greater formalism.

Preliminary results have already shown that the lexical similarity decreases as the levels are farther away from each other in the scale.

4 Expected Results and Further Work

Our aim with this interactive platform is to demonstrate proof of concept for a selection of known combinatorial problems. The goal is to have the LLM output a natural language description of the reasoning behind the solver's search for a solution. It would alternate between stepwise explanation, explaining the deductions made along the way, and counterfactual or MUS-based explanations [6] to explain dead ends in the search tree. The platform would also dynamically adapt to the expertise of the user for a better experience.

Our planned experiments include testing the potential for improvement of the LLM translations when documentation is provided to the LLM using a Retrieval Augmented Generation (RAG) module.

One of the next steps to close the loop would be to also use automatic modeling tools integrated into the chat via an MCP server. This way, users would get access to a fully automatic tool to help them model, solve, and explain the findings of a solution to their model.

References

1 Bart Bogaerts, Emilio Gamba, Tias Guns, and Jens Claes. Step-wise explanations of constraint satisfaction problems. In *European Conference on Artificial Intelligence (ECAI 2020)*, pages 640–647. IOS Press, 2020.

2 Augusto B Corrêa, André G Pereira, and Jendrik Seipp. Classical planning with llm-generated heuristics: Challenging the state of the art with python code. *Advances in Neural Information Processing Systems*, 38:42070–42108, 2026.

3 Augustin Crespin, Ioannis Kostis, Hélène Verhaeghe, and Pierre Schaus. Cp-model-zoo: A natural language query system for constraint programming models. *arXiv preprint arXiv:2509.07867*, 2025.

4 Vikas Hassija, Vinay Chamola, Atmesh Mahapatra, Abhinandan Singal, Divyansh Goel, Kaizhu Huang, Simone Scardapane, Indro Spinelli, Mufti Mahmud, and Amir Hussain. Interpreting black-box models: a review on explainable artificial intelligence. *Cognitive Computation*, 16(1):45–74, 2024.

5 Xinyi Hou, Yanjie Zhao, Shenao Wang, and Haoyu Wang. Model context protocol (mcp): Landscape, security threats, and future research directions. *arXiv preprint arXiv:2503.23278*, 2025.

6 Alexey Ignatiev, Alessandro Previti, Mark Liffiton, and Joao Marques-Silva. Smallest mus extraction with minimal hitting set dualization. In *International Conference on Principles and Practice of Constraint Programming*, pages 173–182. Springer, 2015.

7 Kostis Michailidis, Dimos Tsouros, and Tias Guns. Cp-bench: Evaluating large language models for constraint modelling. *arXiv preprint arXiv:2506.06052*, 2025.

8 Kostis Michailidis, Dimos Tsouros, Tias Guns, and P Shaw. Constraint modelling with llms using in-context learning. In *30th International Conference on Principles and Practice of Constraint Programming (CP 2024)*, volume 307, pages 20–1, 2024.

9 Mahdi Mostajabdaveh, Timothy T Yu, Rindranirina Ramamonjison, Giuseppe Carenini, Zirui Zhou, and Yong Zhang. Optimization modeling and verification from problem specifications using a multi-agent multi-stage llm framework. *INFOR: Information Systems and Operational Research*, 62(4):599–617, 2024.

10 Weichun Shi, Minghao Liu, Wanting Zhang, Langchen Shi, Fuqi Jia, Feifei Ma, and Jian Zhang. Constraintllm: A neuro-symbolic framework for industrial-level constraint programming. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 16010–16030, 2025.

11 Florentina Voboril, Vaidyanathan Peruvemba Ramaswamy, and Stefan Szeider. Streamllm: Enhancing constraint programming with large language model-generated streamliners. In *2025 IEEE/ACM 1st International Workshop on Neuro-Symbolic Software Engineering (NSE)*, pages 17–22. IEEE Computer Society, 2025.

12 Xuan Wu, Di Wang, Chunguo Wu, Lijie Wen, Chunyan Miao, Yubin Xiao, and You Zhou. Efficient heuristics generation for solving combinatorial optimization problems using large language models. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 3228–3239, 2025.