Practical D2T 2024

The 2nd Workshop on Practical LLM-assisted Data-to-Text Generation

Proceedings of the Workshop

September 23, 2024

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
209 N. Eighth Street
Stroudsburg, PA 18360
USA
Tel: +1-570-476-8006
Fax: +1-570-476-0860
acl@aclweb.org

ISBN 979-8-89176-126-1

Preface

We present the Proceedings of The 2nd Workshop on Practical LLM-assisted Data-to-Text (Practical D2T). This year's Practical D2T takes place at INLG 2024 on Sept 23 in Tokyo, Japan. We would like to thank the INLG organisers for their support.

Natural Language Generation (NLG) has been an active area of research for decades, both academically and industrially. Data-to-text (D2T) generation is the NLG task where a system describes structured data in natural language. Traditionally, commercial D2T systems have been based on symbolic approaches, i.e. handcrafted rules or templates. More experimental approaches to D2T, such as E2E and Transformer-based systems have been limited to research because of well-known issues like knowledge gaps, lack of factuality, and hallucination.

The recently introduced instruction-tuned, multi-task Large Language Models (LLMs) promise to become a viable alternative to rule-based D2T systems. They exhibit the ability to capture knowledge, follow instructions, and produce coherent text from various domains. However, even the best LLMs still suffer from well-known issues of neural models, such as lack of controllability and risk of producing harmful text. Recent research thus proposed various approaches to improve the semantic accuracy of LLMs D2T, including prompt tuning, targeted fine-tuning, Retrieval Augmented Generation (RAG), external tool integration, and neuro-symbolic approaches.

Practical D2T 2024 aims to build a space for researchers to discuss and present innovative work on D2T systems using LLMs.

This year, we are excited to present two keynotes covering the use of LLMs in D2T and related tasks. The keynote speakers are:

- Craig Thomson, Dublin City University / ADAPT, UK
- Marco Valentino, Idiap Research Institute, Switzerland

Practical D2T hosts a hackathon (for the second consecutive time), which this year is focused on the evaluation and semantic accuracy of D2T using LLMs. The hackathon will allow participants to explore the challenges of using LLMs for both generating textual summaries of structured data and text span error annotation of them.

Finally, the workshop features a panel of experts on D2T who will discuss the use of LLMs for generating text from data. They will cover the main challenges involved and share insights on the latest developments in this area.

The Practical D2T 2024 program chairs,
Simone Balloccu (lead), Charles University
Zdeněk Kasner, Charles University
Ondřej Plátek, Charles University
Patrícia Schmidtová, Charles University
Kristýna Onderková, Charles University
Mateusz Lango, Charles University
Ondřej Dušek, Charles University
Lucie Flek, University of Bonn
Ehud Reiter, University of Aberdeen
Dimitra Gkatzia, Edinburgh Napier University
Simon Mille, ADAPT Centre

The organisation of Practical D2T 2024 was partially funded by the European Union (ERC, NG-NLG, 101039303).

Organizing Committee

Workshop Chairs

Simone Balloccu (lead), Charles University
Zdeněk Kasner, Charles University
Ondřej Plátek, Charles University
Patrícia Schmidtová, Charles University
Kristýna Onderková, Charles University
Mateusz Lango, Charles University
Ondřej Dušek, Charles University
Lucie Flek, University of Bonn
Ehud Reiter, University of Aberdeen
Dimitra Gkatzia, Edinburgh Napier University
Simon Mille, ADAPT Centre

Table of Contents

Beyond the Hype: Identifying and Analyzing Math Word Problem-Solving Challenges for Large Language Models

Romina Soledad Albornoz-De Luise[1], David Arnau[2], Pablo Arnau-González[1], Miguel Arevalillo-Herráez[1]
[1]Departament d'Informàtica, Universitat de València,
Avinguda de la Universitat s/n, Burjassot, 46100, València, Spain
[2]Departament de Didàctica de la Matemàtica, Universitat de València,
Avinguda dels Tarongers, 4, València, 46022, Spain
{romina.albornoz, david.arnau, pablo.arnau, miguel.arevalillo}@uv.es

Abstract

Despite not being explicitly trained for this purpose, models like Mistral and LLaMA have demonstrated impressive results across numerous tasks, including generating solutions to Mathematical Word Problems (MWPs). A MWP involves translating a textual description into a mathematical model or equation that solving it. However, these models face challenges in accurately interpreting and utilizing the numerical information present in the MWP statements, which can lead to errors in the generated solutions. To better understand the limitations of LLMs, we analyzed the MWP where models failed to accurately solve problems from the SVAMP dataset. By categorizing these MWPs, we identify specific types of problems where the models are most prone to errors, providing insights into the underlying challenges faced by LLMs in problem-solving scenarios and open new modeling opportunities. By understanding the expected errors, researchers can design strategies to adequately model problems more effectively and choose the most suitable LLM for solving them taking into account each model's strengths and weaknesses.

1 Introduction

LLMs have expanded the boundaries of understanding and generating natural language (Karanikolas et al., 2024). Moreover, recent research has found LLMs to be capable of producing high-quality source code (Rozière et al., 2024). LLMs excel at producing text sequences, but also show reasoning capabilities that have been previously applied to Math Word Problem (MWP) Solving (Kojima et al., 2023) by transforming the MWP in natural language to the mathematical language.

In this context, recent research (Arnau-González et al., 2024) has explored LLMs in the context of education by producing source-code that can be compiled into a solution graph for tutoring and supervision purposes.

This paper aims to investigate the types of MWP statements that LLMs have difficulties solving by analyzing incorrect samples produced in previous studies. To this end, we select the SVAMP dataset and three different models: OpenMath/Mistral-7B from Nvidia, Llama3-8B[1], and CodeLlama 34B[2] (Rozière et al., 2024), which demonstrated high performance in MWP-solving task. By focusing on problem statements where these models failed, we identify patterns in the sources of errors. The provided analysis[3] can direct research towards a better understanding of the reasoning limitations of language models.

2 Background and related work

A MWP model solution can be understood as the result of reducing the initial MWP to a graph of mathematical relationships between quantities.

Consider the MWP where we have bought a car and paid 12 bills of 400 euros each. If the car costs 12 800 euros, we need to determine how much money is left to pay. A possible problem model would establish the following relationships: The total price of the car equals the sum of the money already paid and the money left to pay. Furthermore, the money already paid is calculated by multiplying the value of a single bill by the number of bills paid.

Automatically solving a math problem articulated in natural language presents a significant challenge, necessitating both comprehension and accurate reasoning. This process requires techniques to extract not only the quantities explicitly stated in the MWP but also those implied by terms such

[1]https://llama.meta.com/llama3/
[2]https://llama.meta.com/code-llama/
[3]Analysed dataset available in https://zenodo.org/records/12771266

Proceedings of the 2nd Workshop on Practical LLM-assisted Data-to-Text Generation, pages 1–6
September 24, 2024.

as “twice”, “half” or “left”. Additionally, solving the MWP demands an understanding of the relationships among these quantities, the identification of the target quantity, and the sequence of operations needed to achieve the final result. In essence, solving a problem from natural language is a task primarily concerned with knowledge extraction and the identification of advanced relationships (Jie et al., 2022a). Recent studies have shown that LLMs show a problem-solving ability similar to that of children, despite differences in the type of MWP they are able to solve best (Arnau-Blasco et al., 2024).

The task of automated MWP solving has been a topic of interest in the literature since the 1960s, inspiring a recent survey (Zhang et al., 2020). Recent efforts in solving Mathematical Word Problems (MWPs) have concentrated on constructing expression (or equation) trees. These methods focus on creating arithmetic expressions by forming equivalent trees. However, due to the exponential growth of the search space as the number of quantities increases, alternatives leveraging reinforcement learning techniques have been explored (Wang et al., 2018).

In the last year, prompting techniques have been developed to force reasoning on decoder-only transformers (Kojima et al., 2023). Moreover, the introduction of new-generation LLMs like Llama2 and Mistral has also led to new studies in the field. In this direction, (Arnau-González et al., 2024) proposed a method that incorporates MWP solving, quantity value assignment and naming as well as capabilities for establishing relationships, without the need for fine-tuning the underlying LLM.

3 Dataset

The SVAMP MWP dataset (Patel et al., 2021) consists of 1000 elementary-level arithmetic word problems, each solvable by expressions requiring no more than two operators. This dataset provides annotated solutions for each MWP. SVAMP was selected for evaluating and analyzing the performance of LLMs in solving MWPs, being widely recognized as one of the most challenging datasets for arithmetic MWP solving (Patel et al., 2021; Jie et al., 2022b).

3.1 Studied samples

In a previous study (Arnau-González et al., 2024), the authors have developed a method where a LLM is prompted with an example MWP statement alongside a corresponding correct Python function named sol(), which solves the MWP. Additionally, the model receives the MWP statement to be solved and a partially defined Python function. The model then completes the function by defining quantities and their relationships, and returning the requested result. Figure 1 shows this process with annotations highlighting the different parts of the prompt and the generated output. The accuracy of the solution can be verified by executing the generated Python code and comparing its result to the expected solution.

Prompt

Example

```python
""" A book has 3 chapters. The first chapter is 91 pages
long the second chapter is 23 pages long and the third
chapter is 25 pages long. How many more pages does the
first chapter have than the second chapter? """

def sol():
    context = dict()
    context['number of chapters'] = 3
    context['number of pages first chapter'] = 91
    context['number of pages second chapter'] = 23
    context['number of pages third chapter'] = 25
    context['pages more first chapter'] =
    (context['number of pages first chapter']
    - context['number of pages second chapter'])
    return context['pages more first chapter']
```

Input Problem

```python
""" Each pack of dvds costs 76 dollars.
If there is a discount of 25 dollars on each pack.
How much do you have to pay to buy each pack? """

def sol():
    context = dict()
```

Generated output

```python
context['price per pack'] = 76
context['discount per pack'] = 25
context['price after discount'] =
context['price per pack'] -
context['discount per pack']
return context['price after discount']
```

Figure 1: Python code with generated output example: Prompts in yellow sections and an example of LLM generated output in green.

The samples provided in (Arnau-González et al., 2024) are an attempt to automatically solve MWP using Python code.

Originally, the published samples contained Python source code produced by 19 different LLMs, on the problems contained in the SVAMP and GSM-8k datasets. Each model was used to generate 10 independent solutions for each MWP for three different temperature settings $\tau \in \{0.1, 0.3, 0.5\}$. In this study, we focus our analysis on the top-3 top-performing models based on the best accuracy values, computed considering only the first of the 10 solutions generated. These models were OpenMath-Mistral-7b, Llama3-8b, and Codellama-34b, when using $\tau = 0.1$.

4 Analysis

We focused our analysis on the characteristics of MWPs that LLMs tend to solve incorrectly. These

MWPs have been selected by studying the generated samples obtained in previous studies (Arnau-González et al., 2024). Out of the three studied models we have chosen CodeLlama-34b as the worst model, as it provided the worst accuracy on the SVAMP dataset.

We decided to delve into those problems that the worst performing model (CodeLLama-34b) failed to solve accurately, in order to draw some conclusions on the structure of said MWPs.

After an initial visual inspection, we found specific patterns in the MWPs for which models tend to fail and categorized them to better understand the limitations of the models based on the type of MWP they are trying to solve, into at least one of the following three types: MWP with unfeasible solutions (US), MWP with unnecessary quantities (UQ), and MWP involving comparisons (CP).

MWPs that fall into the Unfeasible Solution category are MWPs that, although can be solved analytically, the solution obtained is not practical or possible in the presented scenario. This happens, for instance, when one or more quantities in the solution take an integer or real value where only natural numbers are physically possible. An example of this type of statement is *"A waiter had 12 customers. After some left he still had 14 customers. Then he got 10 new customers. How many customers does he have now?"*. This problem implies that somehow -2 customers left the restaurant.

MWPs with the Unnecessary Quantities category contain one or more quantities that are not required to solve the problem. We have observed that most LLMs have a natural tendency to use all the quantities present in the statement to produce a solution, typically leading to errors in the reasoning. A good example of these MWP is *"Rebecca wants to split a collection of eggs into groups of 6. Rebecca has 18 eggs 72 bananas and 66 marbles. How many groups will be created?"*. In this case, bananas and marbles amounts are unnecessary in determining how many groups of eggs will be created. However, models will still add it to the number of eggs.

What does the Table 2 with Mann-Whitney U rank test show? In the table are the statistics? Finally, MWPs falling in the Involving a Comparison category, have their statements asking or involve a direct comparison between two quantities. This often confuses models as sometimes are not capable of capturing these relationships appropriately. An example of this type of problem is *"There were 3 dollars in Olivia's wallet. She collected 49 more dollars from an ATM. After she visited a supermarket there were 49 dollars left. How much more money did she collect at the ATM than she spent at the supermarket?"*. In this statement, the problem question is a comparison between money collected at the ATM and the amount spent at a supermarket. Table 1 summarizes the types of MWP statements along with brief descriptions of each category.

Other categories have been created by combining the previously identified categories, US$\wedge$UQ, US$\wedge$CP, UQ$\wedge$CP, and US$\wedge$UQ$\wedge$CP. These categories contain the samples that can be tagged in more than one category. Finally, an additional category UNIDENTIFIED has been created, containing the MWPs that have not been labelled in any of the previous categories. This type of problems has a simple structure with no irrelevant comparison elements or quantities.

MWPs that CodeLlama-34B fails to solve correctly have been classified by two independent volunteer annotators. The annotators were asked to tag which of the identified problems were present in each of the selected MWPs. According to both annotators' responses we computed Cohen's Kappa for each of the three separate problems. In all cases, we obtained a $\kappa > 0.6$, indicating substantial agreement was achieved by the annotators. Finally, since annotator #1 had more experience in the field, it was decided to choose samples from that annotator. The analysis of the classification of the selected samples shows that the MWPs can be classified into at least one of the identified categories in over 70% of the MWPs which CodeLlama fails to solve.

CodeLlama fails to solve 45.8% of problems in the UQ category, indicating that these problems present the greatest challenge for the model. 37.8% in the CP category, and 21.8% in the US category. Additionally, 14.5% of the problems fall into both the US and UQ categories, 10.9% fall into both the US and CP categories, 16% fall into both the UQ and CP categories, and 6.9% fall into all three categories: US, UQ, and CP. Finally, 29.1% of the problems do not fall into any of these specified categories. An intriguing observation arises when examining this category of the unsolved problems. These problems typically have clearer statements and generally require a straightforward operation to reach a solution. Despite their apparent simplicity, CodeLlama still faces notable challenges in solving these problems.

In summary, a total of 275 MWPs (those which CodeLlama failed to solve) were selected for anal-

Table 1: Features of MWPs Statements and Brief Description

Category of MWP Statements	**Description**
Unfeasible Solution (US)	MWPs that can be solved analytically, but the solution obtained is not possible in the presented scenario
Unnecessary Quantities (UQ)	MWP statements contain one or more quantities that are not required to solve the problem
Involving a Comparison (CP)	MWP have statements asking or involve a direct comparison between two quantities
UNIDENTIFIED	MWP statements in this category do not exhibit any of the previously mentioned characteristics

Table 2: Category-Wise MWP Statements where Mistral and Llama Models fail Focused on CodeLlama Failures.

	US	**UQ**	**CP**	**US∧UQ**	**US∧CP**	**UQ∧CP**	**US∧UQ∧CP**	**UNIDENTIFIED**
Mistral	43.33	45.24	36.54	40.00	33.33	31.82	26.32	22.50
Llama	61.67	50.79	62.5	50.00	66.67	45.45	47.37	36.25
Total	60	126	104	40	30	44	19	80

The table presents the percentage of problems within each category that are incorrectly resolved by Mistral and Llama models, focusing on problems that CodeLlama initially failed to solve. Additionally, it includes the number of MWPs identified within each category.

Table 3: Mann-Whitney U rank test

	US	**UQ**	**CP**
Mistral	0.0044	0.0004	0.0204
Llama	0.0014	0.0207	0.0002

P-values from the Mann-Whitney U rank test comparing error rates between problems categorized as US, UQ, and CP versus those in the UNIDENTIFIED category. Each sample consists of independent sets of problems, where the identified categories (US, CP, UQ) are compared against the UNIDENTIFIED problems to assess differences in error rates.

ysis in the top two top-performing models. Table 2 displays the error rates for both Mistral and Llama-3 models for the tagged samples, and all the possible tag combinations. A first analysis reveals that the identified categories are indeed problematic also for these models. This is shown by the error rate being higher in all the identified categories and their combinations.

This observation is further supported by a hypothesis test, where the alternative hypothesis (H_a) posits that the distribution of error rates for each identified category (UQ, US, CP) is significantly higher than that of errors in the UNIDENTIFIED category. Specifically, we hypothesize that the error rate in problems identified with UQ (104 problems), US (60 problems), and CP (126 problems) is greater than the error rate in the UNIDENTIFIED category (80 problems). The tested distributions are independent, and since the assumptions of the Students's test are violated (homoscedasticity and normality of the date), we choose a non-parametric alternative. The null hypothesis (H_0) for each comparison is that there is no difference between the error distributions of the identified category and the UNIDENTIFIED category. In other words, H_0 asserts that the two distributions have the same median error rate. The tested distributions represent the proportion of problems solved correctly (is_correct) within each category. These distributions are independent and consist of binary outcomes (True/False). The results, as shown in Table 3, indicate that for all identified categories (UQ, US, CP), there are significantly more errors compared to the UNIDENTIFIED category. This supports our hypothesis that the identified categories represent problem types that are systematically more challenging for the models.

Within the 275 problems incorrectly solved by CodeLlama, we analyzed the errors made by the Mistral and Llama models. Mistral accounted for errors in 173 problems, with 60% of these errors being attributable to the same issues present in the CodeLlama model. Similarly, Llama had 247 problems with erroneous solutions, and 58% of these errors could be explained by the same mistakes made by CodeLlama. This indicates that we were able to identify the characteristics of MWPs of 60% of Mistral's errors and 58% of Llama's errors, respectively. Through this analysis, we can understand a significant portion of the common sources of errors in both Mistral and Llama. In Table 2, we analyze the percentage distribution of incorrectly resolved problems across each category.

5 Conclusion and Future Work

In this work, we have analyzed the performance of three LLMs in solving MWPs on the SVAMP dataset and categorized the sources of errors. The categorization of MWP statement error sources reflects specific patterns in which the models fail to correctly solve these problems. Identifying these patterns provides valuable insights into the limitations of current LLMs.

The provided study shows that, in general, there are three categories of challenging problems for which models tend to generate wrong solutions. Moreover, the results show that statistically, models tend to fail significantly more in problems that fall into one of these categories than in any other type of problems. The fact that these categories can be identified, and the shown difference in performance in these categories shows that LLMs are still weak for certain types of reasoning. The identification of the reasons that cause an incorrect solution in the case of statements that cannot be classified in any of the 3 identified categories is not straightforward. The initial inspection showed that these problems apparently have a clear statement, and there is no reason as to why the models consistently fail to solve the problem. A possible explanation of this issue might be related to the type of relations encoded in the MWPs, as suggested by previous research (Arnau-Blasco et al., 2024).

The presented analysis is a work in progress which examines the characteristics of MWP statements where the selected LLMs fail to provide correct solutions. The initial categorical classification offered as part of this work serves as a preliminary step towards modeling math problems based on categories that reflect the likelihood of being correctly solved by different LLMs. Future work will continue analysing the samples for which the top-performing models fail, in order to gather insights into the reasoning gaps and generate strategies to overcome such failures. We also plan to examine and compare the error rates across different categories made by LLMs with those made by real students.

Acknowledgements

This work has been supported by project PID2023-150960NB-I00, funded by the Spanish Ministry of Science, Innovation and Universities and the European Union; project CIGE/2023/063 and grant CIAPOS/2022/063 funded by Generalitat Valenciana; project TED2021-129485B-C42 funded by MCIN/AEI/10.13039/501100011033 and the European Union "NextGenerationEU"/PRTR; and grant PRE2019-090854, funded by MCIN/AEI/10.13039/ 501100011033 and "ESF Investing in your future".

References

Jaime Arnau-Blasco, Miguel Arevalillo-Herráez, Sergi Solera-Monforte, and Yuyan Wu. 2024. Using large language models to support teaching and learning of word problem solving in tutoring systems. In *Generative Intelligence and Intelligent Tutoring Systems*, pages 3–13, Cham. Springer Nature Switzerland.

Pablo Arnau-González, Stamos Katsigiannis, Ana Serrano-Mamolar, and Miguel Arevalillo-Herráez. 2024. Result outputs for "Automated Math Word Problem solving and quantity identification using Large Language Models for code synthesis".

Zhanming Jie, Jierui Li, and Wei Lu. 2022a. Learning to reason deductively: Math word problem solving as complex relation extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5944–5955, Dublin, Ireland. Association for Computational Linguistics.

Zhanming Jie, Jierui Li, and Wei Lu. 2022b. Learning to reason deductively: Math word problem solving as complex relation extraction. *Preprint*, arXiv:2203.10316.

Nikitas Karanikolas, Eirini Manga, Nikoletta Samaridi, Eleni Tousidou, and Michael Vassilakopoulos. 2024. Large language models versus natural language understanding and generation. In *Proceedings of the 27th Pan-Hellenic Conference on Progress in Computing and Informatics*, PCI '23, page 278–290, New York, NY, USA. Association for Computing Machinery.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large language models are zero-shot reasoners. *Preprint*, arXiv:2205.11916.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2080–2094, Online. Association for Computational Linguistics.

Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, and Gabriel Synnaeve. 2024. Code llama: Open foundation models for code. *Preprint*, arXiv:2308.12950.

Lei Wang, Dongxiang Zhang, Lianli Gao, Jingkuan Song, Long Guo, and Heng Tao Shen. 2018. Mathdqn: Solving arithmetic word problems via deep reinforcement learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Dongxiang Zhang, Lei Wang, Luming Zhang, Bing Tian Dai, and Heng Tao Shen. 2020. The gap of semantic parsing: A survey on automatic math word problem solvers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(9):2287–2305.

Enhancing Situation Awareness through Model-Based Explanation Generation

Konstantinos Gavriilidis[1], Ioannis Konstas[1], Helen Hastie[2], Andrea Munafo[3], Wei Pang[1]
[1]School of Mathematical and Computer Sciences, Heriot-Watt University, Edinburgh, UK
[2]School of Informatics, University of Edinburgh, Edinburgh, UK
[3]SeeByte Limited, Edinburgh, UK
[1] `{kg47, I.Konstas, W.Pang}@hw.ac.uk`
[2] `h.hastie@ed.ac.uk` [3] `andrea.munafo@seebyte.com`

Abstract

Robots are often deployed in remote locations for tasks such as exploration, where users cannot directly perceive the agent and its environment. For Human-In-The-Loop applications, operators must have a comprehensive understanding of the robot's current state and its environment to take necessary actions and effectively assist the agent. In this work, we compare different explanation styles to determine the most effective way to convey real-time updates to users. Additionally, we formulate these explanation styles as separate fine-tuning tasks and assess the effectiveness of large language models in delivering in-mission updates to maintain situation awareness. The code and dataset for this work are available at: `https://github.com/konsgavriiil/explainable_robotics_lm`.

1 Introduction

Automation offers significant advantages in our society, particularly in critical sectors like manufacturing and offshore applications, as recognized in prior studies (Ballestar et al., 2021; Khalid et al., 2022). Fostering transparency and accountability within robotics is imperative to bolster trust and wider adoption (Wachter et al., 2017; Winfield et al., 2021). One pivotal cognitive process influencing trust and adoption is situational awareness, characterized by three essential stages: **perception** (understanding a robot's decision-making), **comprehension** (discerning the rationale behind these decisions), and **projection** (anticipating future automated behaviours). Recent research has shown that textual explanations presented visually within Human-In-The-Loop applications, such as autonomous driving, positively impact all facets of situational awareness (Avetisyan et al., 2022).

In this work, we include two user studies focusing on situation awareness and explanation generation. We share a dataset, licensed under Creative Commons Attribution (CC-BY), which contains categorical events related to maritime autonomous missions, user queries, and corresponding explanations. Additionally, we demonstrate the performance of multiple large language models on three downstream tasks derived from our dataset.

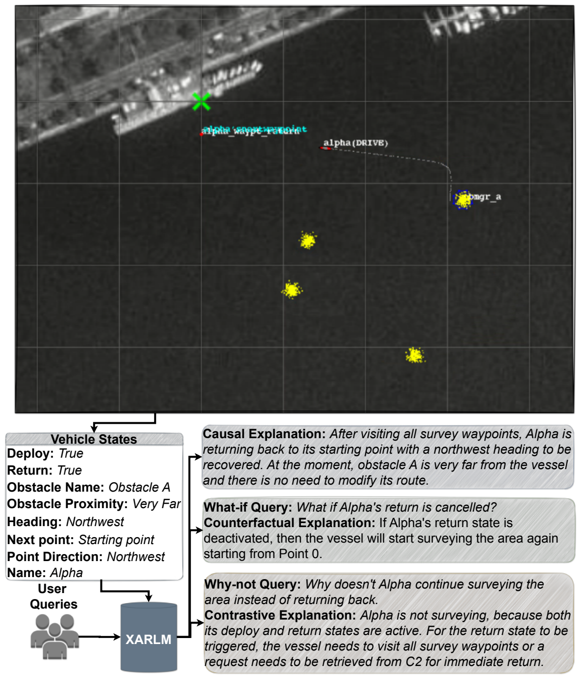


Figure 1: The eXplainable Autonomous Robot Language Model (XARLM) retrieves vehicle states and user queries to generate explanations in various styles, thereby enhancing situation awareness for vehicle operators.

Through the fine-tuning process and the user studies, we aim to answer the following research questions:

- **RQ1:** How robust are large language models in delivering explanations of autonomous mission events in causal, counterfactual, and contrastive styles?
- **RQ2:** Which of the three explanation styles

Proceedings of the 2nd Workshop on Practical LLM-assisted Data-to-Text Generation, pages 7–16

most effectively enhances situation awareness for users?

- **RQ3:** Do users prefer model-based explanations over template-based explanations?

The remainder of this paper is structured as follows: Section 2 reviews prior research that has influenced our approach. Section 3 describes the data collection and annotation processes. Section 4 outlines the fine-tuning process for the large language models and details the experiments conducted to identify the best-performing model. In Section 5, we describe the tasks included in our study to address research questions 2 and 3, as well as the participant groups that completed the questionnaire. Section 6 presents the performance of the large language models and our findings from the user studies. Finally, Section 7 examines the implications of our findings, and Section 8 discusses potential future experiments and concludes the paper.

2 Related Work

Explainable agents and robots have become a crucial research area due to the increasing demand for transparency and interpretability in autonomous systems (Langley et al., 2017; Anjomshoae et al., 2019). These systems must effectively communicate their decision-making processes to users, preferably through user-friendly modalities such as natural language (Cambria et al., 2023). Typically, natural language explanations are presented as causal explanations, which are easy to understand and clearly justify automated behaviours (Diehl and Ramirez-Amaro, 2022). Other types of explanations, such as counterfactual explanations (answering "What if" questions) and contrastive explanations (answering "Why not" questions), also facilitate the interrogation of black-box systems (Stepin et al., 2021).

Generating these explanations faithfully involves sophisticated methods for content selection, such as using Bayesian networks or surrogate models (Gyevnar et al., 2022; Gavriilidis et al., 2023). The selected content can then be communicated through controllable template-based approaches (Hastie et al., 2017). Additionally, end-to-end approaches using encoder-decoder architectures have shown promise in conveying agent rationale and improving failure and solution identification (Ehsan et al., 2019; Das et al., 2021).

The advent of causal language models with transformer-based encoder architectures (Touvron et al., 2023; Jiang et al., 2023) has significantly advanced the field of text generation. These models excel at replicating domain-specific knowledge due to extensive training on vast amounts of human-generated text (Kıcıman et al., 2023). Despite their substantial size and complexity, new techniques such as QLoRA (Dettmers et al., 2024) have made fine-tuning more computationally efficient, facilitating the adaptation of pre-trained models to specific downstream tasks.

To evaluate the semantic accuracy of models, researchers frequently compare the outputs with their corresponding inputs (Xu et al., 2021). This evaluation approach is particularly important in applications where the accuracy and reliability of natural language explanations are critical. Commonly used metrics for this purpose include BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), and METEOR (Banerjee and Lavie, 2005), which measure quality based on n-gram overlap between reference labels and model-generated responses. However, these metrics have limitations, as they often fail to capture the true semantic similarity or desired verbosity of the outputs (Zhao et al., 2020). To address these shortcomings, combining n-gram-based metrics with additional metrics that perform verbatim comparisons can provide a more comprehensive evaluation of model outputs, particularly in high-stakes applications.

Given the robustness of large language models in data-to-text generation and their capability to perform multiple tasks, various domains have leveraged these models for diverse applications. For instance, they have been used for action selection in embodied tasks (Ahn et al., 2022) and for text summarization to infer sets of rules for object manipulation based on user preferences (Wu et al., 2023). In the realm of explainable robotics, language models are combined with Retrieval Augmented Generation (Lewis et al., 2020) to transform robot logs and user queries into natural language explanations, thereby enhancing human-robot interaction.

3 Data Generation

To collect a dataset for autonomous maritime vehicles, we deployed an agent that follows a pre-existing plan and attempts to complete its objectives by visiting a set of waypoints using different patterns (e.g., lawnmower, loiter). The agent prioritizes the integrity of the robotic platform and replans its behaviour in case of unexpected events. At

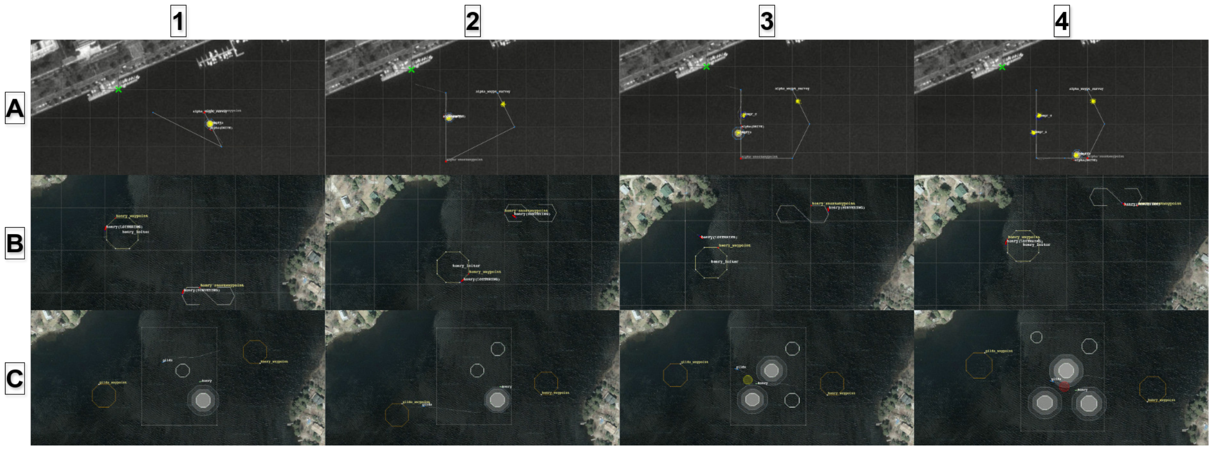


Figure 2: The MOOS-IvP scenarios used for data generation. Each of the three scenarios includes four different configurations with varying waypoints and objectives.

each simulation timestep, we recorded the robot's behaviour and the states affecting that behaviour, such as objectives and sensor-derived events (e.g., obstacle or vessel detection).

For this dataset, we utilised MOOS-IvP, an open-source behavioural agent designed for maritime robots (Benjamin et al., 2010). This simulator offers a variety of pre-built scenarios, from which we selected and refined three specific missions. We further modified the mission plans, creating four distinct configurations for each scenario. We limited the logged vehicle states to those impacting the agent's behaviour activations. Finally, we performed post-mission parsing of the log files to extract the relevant vehicle states and the corresponding activated behaviours.

In Figure 2, we illustrate three scenarios, each with four distinct task and environment configurations. Scenario **A** involves an unmanned surface vehicle (USV) conducting a survey, avoiding obstacles, and returning to its starting point for retrieval. Scenario **B** features an unmanned underwater vehicle (AUV) loitering around predefined waypoints and transitioning to a designated survey area upon receiving instructions. Scenario **C** has two vehicles loitering around a random polygon, occasionally switching sides and restarting their routine while avoiding collisions and obstacles. With each scenario, the task difficulty increases by adding more behaviours and introducing complex tasks such as collision avoidance.

3.1 Data Annotation

After completing data collection, model-based data annotation was conducted. Using a larger model,

	C	**CF**	**CT**
Dataset size	1151	3450	3450
Vocabulary size	758	993	1167
Avg Input Length	109.70	121.90	125.22
Longest Input Length	132	153	165
Shortest Input Length	86	96	97
Avg Output Length	42.38	37.13	61.41
Longest Output Length	89	122	151
Shortest Output Length	16	7	21
Inputs with spatial tokens	1151	3450	3450
Avg spatial tokens/input	18.88	22.94	22.06
Outputs with spatial tokens	1146	3128	3379
Avg spatial tokens/output	8.58	6.23	7.83

Table 1: Dataset Statistics for causal (C), counterfactual (CF) and contrastive (CT) explanations.

new annotations were generated for each data instance by providing a small number of instruction-based examples, as guided by prior research (Taori et al., 2023). Specifically, the OpenAI API's Chat-Completion functionality with the GPT-3.5-Turbo model was utilised. Task instructions and concatenated vehicle state representations were input, resulting in potential user queries along with their corresponding explanations. For counterfactual and contrastive explanations, a state or behaviour permutation was also provided, depending on the task, to validate the user query upon which the explanation was based.

Initially, 12 instructions were defined for Scenario A, 15 for Scenario B, and 21 for Scenario C, ensuring that all unique states and behaviours relevant to each scenario were addressed. This process produced an annotated dataset comprising 8,051 data instances, reflecting the state updates encountered during each mission to minimise repeated state-behaviour combinations. Detailed statistics

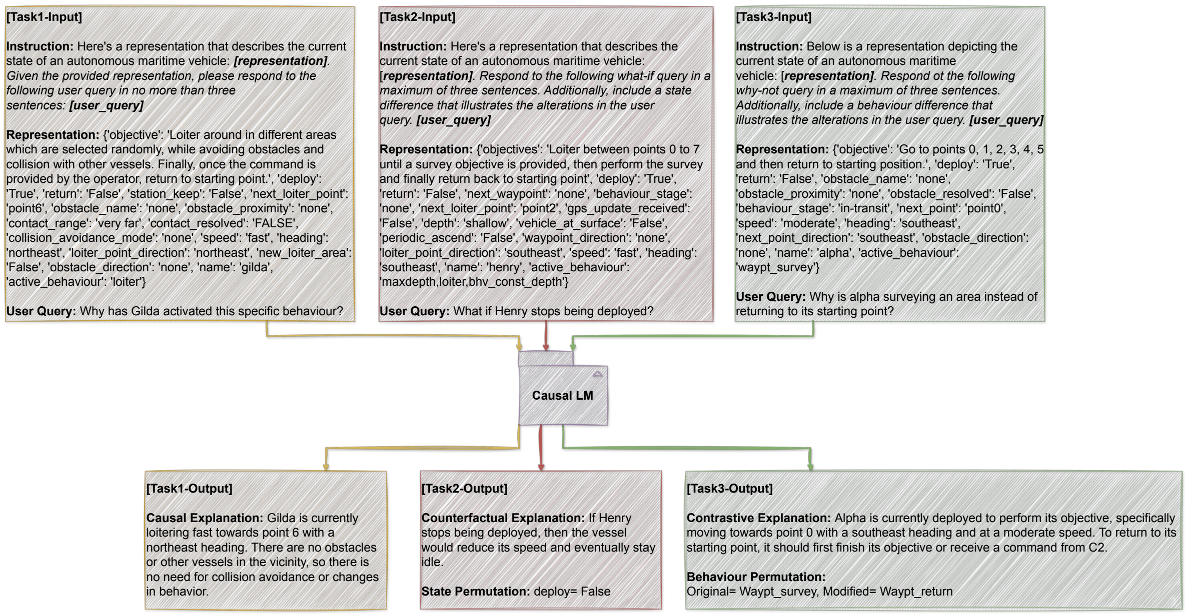


Figure 3: The defined fine-tuning tasks involve a causal language model that retrieves an instruction, a vehicle state representation, and a user query. The model then outputs an explanation, and for counterfactual and contrastive explanations, it additionally provides a permutation.

of the annotated dataset, including vocabulary size, input/output lengths, and the number of spatial tokens, are provided in Table 1.

4 Fine-Tuning

Before attempting any fine-tuning, we evaluated the performance of existing instruction fine-tuned large language models to assess their capability in generating explanations from autonomous vehicle states. Specifically, we employed three transformer-based decoder models, each with an identical number of parameters: Llama2-7B-Chat, Mistral-7B-IT, and Falcon-7B-IT, using 2-shot inference (Touvron et al., 2023; Jiang et al., 2023; Almazrouei et al., 2023). This preliminary experiment revealed that all three models demonstrated strong results in terms of semantic accuracy and precision, with Mistral and Llama2 slightly outperforming Falcon in these aspects. However, when evaluated using machine translation metrics, all three models exhibited significant shortcomings, with Mistral performing slightly better.

Upon further inspection of the model outputs, we found that these models often compensated by increasing verbosity and adding supplementary tokens. These additions were unnecessary and may increase the cognitive load for users reading the explanations. The higher semantic accuracy and precision scores can be attributed to our metric's focus on the output mentioning spatial elements, behaviours, and entities present in the input representation. In contrast, the lower scores in ROUGE-L, BLEU, and METEOR metrics were due to the generated outputs not closely resembling the dataset labels. The results of this initial experiment are illustrated in Figure 4.

Recognizing the need for further refinement for

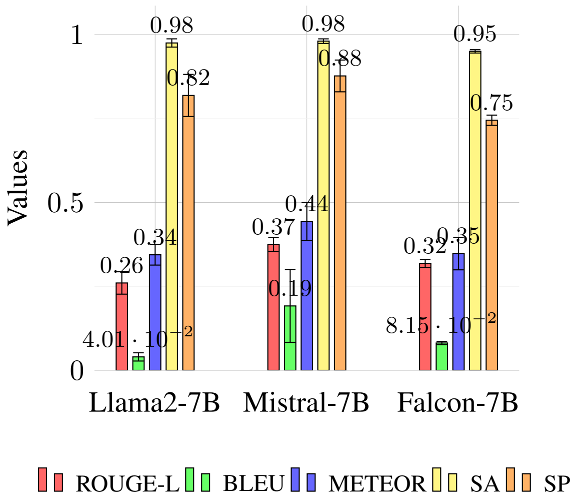


Figure 4: Performance of instruction fine-tuned large language models on two-shot inference tasks, with error bars indicating the mean and variability across various explanation types. The metrics include Semantic Accuracy (SA) and Semantic Precision (SP).

our downstream task, we defined a fine-tuning setup where each explanation type is treated as a separate task. In Figure 3, we represent our fine-tuning tasks, where a task instruction, along with a representation and a user query, are provided as input. The model output is the corresponding explanation, including permutations for counterfactual and contrastive explanations. Using our annotated dataset, we trained the three large language models on all explanation tasks utilizing the HuggingFace and PEFT (Xu et al., 2023) libraries.

4.1 Automatic Evaluation

To evaluate the performance of our models on downstream tasks, three machine translation metrics—BLEU, METEOR, and ROUGE—were utilised to measure n-gram overlap between the model outputs and reference labels. Additionally, to accurately assess the mentions of entities, landmarks, and specific details such as vessel heading, depth, speed, or behaviour, a semantic accuracy and precision metric was developed. The SA metric increases with each correct mention and decreases when elements are inaccurately identified (e.g., using 'medium' instead of 'fast' speed), ensuring the fidelity of the generated explanations.

Specifically, given a set of input tokens I and a set of output tokens O, for each token category (spatial, state, decision) that is based on a vocabulary we predefined, the sets of correct references, true positives, and false positives are defined as follows:

$$\text{Correct References} = I \cap O$$

The number of true positives (TP) and false positives (FP) are calculated as:

$$TP = |\text{Correct References}|$$

$$FP = \text{Total References} - TP$$

The semantic accuracy (Acc) and precision (Prec) are defined by:

$$\text{Acc} = \frac{TP + TN}{|O|}$$

$$\text{Prec} = \frac{TP}{TP + FP}$$

where TN (true negatives) denotes the number of tokens in O that are not references. The overall semantic accuracy and precision are computed as the average across all evaluated references within the spatial, state, and decision categories.

Section 6 presents a performance comparison of the three language models to identify the best-performing model. An ablation study was subsequently conducted on the top-performing model to explore potential improvements.

5 User Study

To estimate the effect of explanations on users and determine user preference for model-based explanations, we designed two user studies. A total of 21 participants were recruited from the robotics industry and academia, including 9 individuals very familiar with autonomous vehicles, 9 who were familiar, and 3 who were not familiar.

User Study on Situation Awareness: This study builds upon prior work (Robb et al., 2018) to investigate the effect of different explanation styles on users' situation awareness. We used recorded videos from the aforementioned maritime robot simulator, where an agent attempts to accomplish a set of objectives while considering its environment and inner state, particularly during unexpected events that require replanning. Participants encountered three different conditions, each with a different explanation type (causal, counterfactual, and contrastive), along with a tutorial video describing the task beforehand. In the first condition, explanations were presented with captions. In the second and third conditions, participants selected user queries to generate corresponding explanations that clarified alternative outcomes. After the explanations were displayed, the interface asked users about events taking place in the video at predefined timesteps. Their responses were used to estimate a performance metric representing their situation awareness per condition, thus assessing the effect of each explanation style on their mental models.

User Study on Explanation Preference: This study presented three separate scenarios, each with a map displaying the vessel and its environment, a description summarizing the events, a user query, and three potential explanations. Two default options allowed users to select all or none of the explanations to avoid restricting their choices. For each scenario, participants chose the explanation that best conveyed the current state of the robot. These explanations were derived from both domain expert templates (with low soundness and high complete-

	ROUGE-L	BLEU	METEOR	Semantic Accuracy	Semantic Precision
Causal	0.631	0.460	0.651	0.978	0.884
Counterfactual	0.665	0.538	0.670	0.969	0.857
Contrastive	0.652	0.561	0.669	0.983	0.902
All types	0.430	0.417	0.459	0.975	0.887

Table 2: Performance comparison of the top-performing large language model, Mistral, on individual tasks as well as on a combined dataset of all three tasks using a balanced dataset.

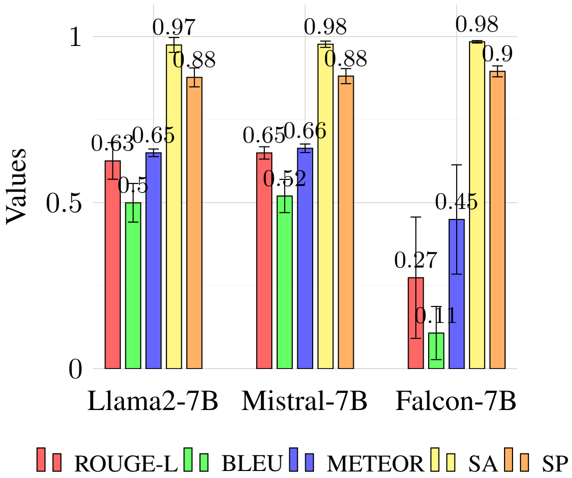


Figure 5: Performance of fine-tuned large language models, showing improved machine translation metrics compared to Figure 4.

ness) and language models, though participants were not informed of their origin. Selections were made based solely on the clarity and informativeness of the explanations provided.

6 Results

In this section, we present the results of our fine-tuned models and user study, addressing the research questions outlined in Section 1.

6.1 Automatic Evaluation

To address **RQ1**, we present the overall performance of the three large language models on the three downstream tasks, as illustrated in Figure 5. Based on the performance metrics, Mistral and Llama2 demonstrated the best results, with Mistral showing a slight edge and a significant improvement in machine translation metrics. These models also achieved high scores in Semantic Accuracy and Precision, indicating that their outputs accurately reflected the vehicle state representations provided as input.

In contrast, the Falcon model performed well on causal explanations but did not achieve comparable performance on the other two explanation types, affecting its mean performance across all tasks. Similar to its behaviour in the instruction version, the Falcon model produced verbose outputs that mixed relevant tokens with supplementary, unnecessary information. These results were evaluated for both the fine-tuned and instruction models using a test set of 100 data instances for each explanation type.

After identifying Mistral-7B as our best-performing model, we evaluated its performance on three individual datasets and a balanced dataset with equal numbers of all explanation types. As shown in Table 2, the model trained on the counterfactual dataset achieved the highest ROUGE-L and METEOR scores. The model trained on the contrastive dataset achieved the best BLEU score. The causal dataset model ranked third in machine translation metrics, with the balanced dataset model coming in last. For semantic accuracy and precision, the contrastive dataset model performed the best, while the causal and balanced dataset models had similar results. The counterfactual dataset model ranked last in semantic accuracy and precision, but not significantly behind the top models.

6.2 User Study

With the results from the two user studies, we address **RQ2** and **RQ3** as outlined in Section 1.

In the first user study, illustrated in Figure 6, we measured the total number of correct answers per condition (causal, counterfactual, and contrastive) and compared these results to the probability of randomly selecting the correct answer (33.3%) to determine the impact of explanations on situation awareness. Causal explanations led to the highest percentage of correct answers (76.19%), followed by contrastive explanations (69.84%) and counterfactual explanations (59.67%). The performance difference between random selection and explanation-assisted answers demonstrates that our explanations enhanced users' ability to correctly perceive events.

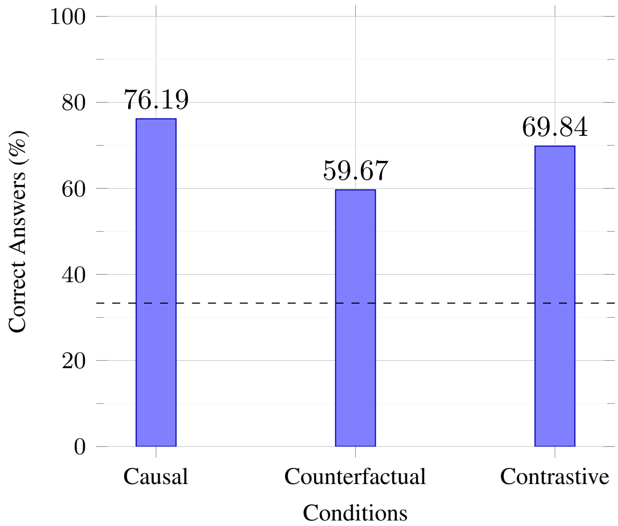


Figure 6: Percentage of correct answers for each condition in the first user study examining the impact of explanation styles on situation awareness.

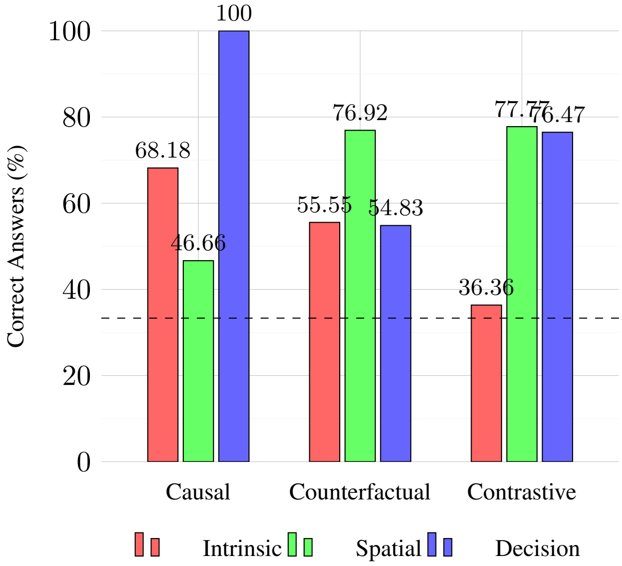
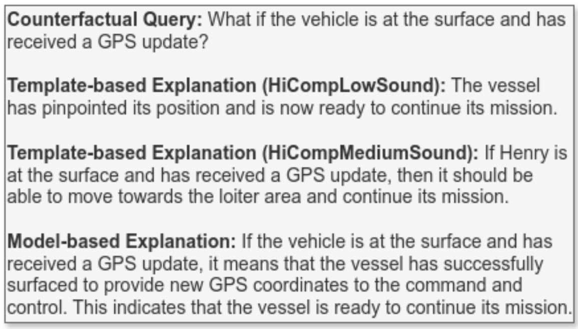


Figure 7: Percentage of correct answers for each condition on questions assessing different aspects of autonomous vehicles (intrinsic states, spatial elements, decision-making).


Counterfactual Query: What if the vehicle is at the surface and has received a GPS update?

Template-based Explanation (HiCompLowSound): The vessel has pinpointed its position and is now ready to continue its mission.

Template-based Explanation (HiCompMediumSound): If Henry is at the surface and has received a GPS update, then it should be able to move towards the loiter area and continue its mission.

Model-based Explanation: If the vehicle is at the surface and has received a GPS update, it means that the vessel has successfully surfaced to provide new GPS coordinates to the command and control. This indicates that the vessel is ready to continue its mission.

Figure 8: Three correct explanations for a counterfactual query, consisting of two template-based explanations with high completeness—one with low soundness and the other with medium soundness.

Further analysis of the first user study involved categorizing the questions into three types: **intrinsic** (inquiring about the robot's internal states, such as sensor readings), **spatial** (concerning the vessel's topology, its environment, and nearby entities or landmarks), and **decision-making** (asking about the rationale behind the robot's decisions). Figure 7 shows that causal explanations resulted in the highest accuracy for intrinsic (68.18%) and decision-making (100%) questions, but the lowest for spatial questions (46.66%), still better than random selection. Counterfactual explanations provided the second-best performance for both intrinsic and spatial questions, showing at least a 20% improvement over random selection. Contrastive explanations led to the best performance for spatial questions (77.77%) and the second-best for decision-making (76.47%), but they performed the worst for intrinsic questions, only slightly better than random selection (36.36%).

In the second study, we explored user preferences between template-based and model-based explanations. Templates created by domain experts, containing only essential information with optimal verbosity, were preferred by 70% of users. Model-based explanations were favored by 15%, while 13.33% liked both types equally, and 1.66% liked none of the explanations. These results suggest that although model-based annotations can accurately depict events, they do not fully match the preferred explanation style of users. This discrepancy indicates that the initial annotation instructions might need refinement to train models that produce explanations more closely aligned with those created by domain experts. Figure 8 presents an example of a what-if query with two template-based explanations and one model-based explanation.

7 Discussion

Our evaluation of inference performance using existing instruction fine-tuned large language models revealed that, despite their inherent capabilities and domain knowledge from pre-training, these models fall short in generating explanations with the verbosity and style expected by domain experts in autonomous vehicles. Consequently, additional fine-tuning on specific downstream tasks is necessary. Our fine-tuned models showed significant improvements in machine translation metrics, indicating a strong n-gram overlap between predictions and reference labels. Notably, our best model performed exceptionally well on counterfactual and contrastive explanations, followed by causal explanations and mixed styles when using a balanced dataset. Furthermore, the generated outputs exhibited high semantic accuracy and precision, underscoring the effectiveness of the fine-tuning process.

The results from the first user study on the effect of different explanation styles on situation awareness demonstrated that users significantly benefited from our explanations compared to random chance, as there were three potential answers per question. Specifically, users gave the most correct answers using causal explanations, followed by contrastive and counterfactual explanations. For causal explanations, users excelled in answering questions about decision-making, as the justification behind the exhibited behaviour was clear and did not require further queries.

Conversely, counterfactual and contrastive explanations allowed users to interrogate the system and learn more about the spatial elements of the mission, resulting in an almost equal percentage of correct answers. While causal explanations helped users answer spatial questions with the third-best success rate, they did not provide enough time to digest the information, potentially increasing cognitive load.

For intrinsic questions concerning the robot's inner states, such as sensor readings, causal explanations demonstrated the best performance, indicating that a straightforward approach to explaining a robot's inner states is the most effective strategy. Considering these findings, future work could tailor the explanation styles presented to users based on the type of content needing explanation.

The results from the second user study indicated a clear preference among participants for domain expert template-based explanations, though some participants preferred model-based explanations, and others expressed no strong preference, showing equal satisfaction with both types. This preference may have been influenced by the presentation format: each scenario featured two template-based explanations characterised by high completeness (the breadth of justifications behind an outcome) and low to medium soundness (the level of detail for each justification), which directly reflected the vessel's current state (Kulesza et al., 2013). In contrast, only one model-based explanation was provided per scenario. The use of model-based data annotation for labelling the dataset may have also impacted the study's outcomes.

Future work should focus on aligning language model outputs more closely with the response styles of domain experts and further refining model-based data annotation techniques, particularly for critical applications. Template-based explanations, while effective, are not scalable, require significant time to develop, and lack robustness, especially when dealing with complex or evolving scenarios. These limitations highlight the need for a data-driven approach using large language models, which offer greater adaptability, efficiency, and the potential to generate contextually relevant explanations at scale.

8 Conclusion and Future Work

This work has successfully demonstrated the impact of different explanation styles on situational awareness across various aspects of a mission, such as decision-making, spatial elements, and inner vehicle states, within the context of human-in-the-loop applications for autonomous vehicles. Additionally, we assessed user preferences between template-based and model-based explanations.

We also showcased the capabilities of our large language model in performing data-to-text tasks, transforming the states of autonomous vehicles into natural language explanations across three different styles. The fine-tuned models have shown satisfactory performance in generating coherent and contextually appropriate explanations.

For future work, several avenues for enhancement and exploration remain open. Experimenting with a broader range of explanation types could provide deeper insights into user preferences and effectiveness. Additionally, integrating additional modalities, such as map or chart-based user interfaces, would be a valuable extension. These interfaces are commonly used in conjunction with autonomous agents and could offer a more comprehensive and interactive explanatory experience for users.

Acknowledgments

This work was funded by the EPSRC CDT in Robotics and Autonomous Systems, SeeByte Ltd, and SRPe. We also extend our gratitude to Dr. Marta Romeo for her support.

References

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. 2022. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, et al. 2023. The falcon series of open language models. *arXiv preprint arXiv:2311.16867*.

Sule Anjomshoae, Amro Najjar, Davide Calvaresi, and Kary Främling. 2019. Explainable agents and robots: Results from a systematic literature review. In *18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019*, pages 1078–1088. International Foundation for Autonomous Agents and Multiagent Systems.

Lilit Avetisyan, Jackie Ayoub, and Feng Zhou. 2022. Investigating explanations in conditional and highly automated driving: The effects of situation awareness and modality. *Transportation research part F: traffic psychology and behaviour*, 89:456–466.

María Teresa Ballestar, Ángel Díaz-Chao, Jorge Sainz, and Joan Torrent-Sellens. 2021. Impact of robotics on manufacturing: A longitudinal machine learning perspective. *Technological Forecasting and Social Change*, 162:120348.

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.

Michael R Benjamin, Henrik Schmidt, Paul M Newman, and John J Leonard. 2010. Nested autonomy for unmanned marine vehicles with moos-ivp. *Journal of Field Robotics*, 27(6):834–875.

Erik Cambria, Lorenzo Malandri, Fabio Mercorio, Mario Mezzanzanica, and Navid Nobani. 2023. A survey on xai and natural language explanations. *Information Processing & Management*, 60(1):103111.

Devleena Das, Siddhartha Banerjee, and Sonia Chernova. 2021. Explainable ai for robot failures: Generating explanations that improve user assistance in fault recovery. In *Proceedings of the 2021 ACM/IEEE international conference on human-robot interaction*, pages 351–360.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.

Maximilian Diehl and Karinne Ramirez-Amaro. 2022. Why did i fail? a causal-based method to find explanations for robot failures. *IEEE Robotics and Automation Letters*, 7(4):8925–8932.

Upol Ehsan, Pradyumna Tambwekar, Larry Chan, Brent Harrison, and Mark O Riedl. 2019. Automated rationale generation: a technique for explainable ai and its effects on human perceptions. In *Proceedings of the 24th international conference on intelligent user interfaces*, pages 263–274.

Konstantinos Gavriilidis, Andrea Munafo, Wei Pang, and Helen Hastie. 2023. A surrogate model framework for explainable autonomous behaviour. *arXiv preprint arXiv:2305.19724*.

Balint Gyevnar, Massimiliano Tamborski, Cheng Wang, Christopher G Lucas, Shay B Cohen, and Stefano V Albrecht. 2022. A human-centric method for generating causal explanations in natural language for autonomous vehicle motion planning. *arXiv preprint arXiv:2206.08783*.

Helen Hastie, Francisco Javier Chiyah Garcia, David A Robb, Pedro Patron, and Atanas Laskov. 2017. Miriam: a multimodal chat-based interface for autonomous systems. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction*, pages 495–496.

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*.

Omer Khalid, Guangbo Hao, Cian Desmond, Hamish Macdonald, Fiona Devoy McAuliffe, Gerard Dooly, and Weifei Hu. 2022. Applications of robotics in floating offshore wind farm operations and maintenance: Literature review and trends. *Wind Energy*, 25(11):1880–1899.

Emre Kıcıman, Robert Ness, Amit Sharma, and Chenhao Tan. 2023. Causal reasoning and large language models: Opening a new frontier for causality. *arXiv preprint arXiv:2305.00050*.

Todd Kulesza, Simone Stumpf, Margaret Burnett, Sherry Yang, Irwin Kwan, and Weng-Keen Wong. 2013. Too much, too little, or just right? ways explanations impact end users' mental models. In *2013 IEEE Symposium on visual languages and human centric computing*, pages 3–10. IEEE.

Pat Langley, Ben Meadows, Mohan Sridharan, and Dongkyu Choi. 2017. Explainable agency for intelligent autonomous systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, pages 4762–4763.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

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.

David A Robb, Francisco J Chiyah Garcia, Atanas Laskov, Xingkun Liu, Pedro Patron, and Helen Hastie. 2018. Keep me in the loop: Increasing operator situation awareness through a conversational multimodal interface. In *Proceedings of the 20th ACM International Conference on Multimodal Interaction*, pages 384–392.

Ilia Stepin, Jose M Alonso, Alejandro Catala, and Martín Pereira-Fariña. 2021. A survey of contrastive and counterfactual explanation generation methods for explainable artificial intelligence. *IEEE Access*, 9:11974–12001.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html*, 3(6):7.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Sandra Wachter, Brent Mittelstadt, and Luciano Floridi. 2017. Transparent, explainable, and accountable ai for robotics. *Science robotics*, 2(6):eaan6080.

Alan FT Winfield, Serena Booth, Louise A Dennis, Takashi Egawa, Helen Hastie, Naomi Jacobs, Roderick I Muttram, Joanna I Olszewska, Fahimeh Rajabiyazdi, Andreas Theodorou, et al. 2021. Ieee p7001: A proposed standard on transparency. *Frontiers in Robotics and AI*, 8:665729.

Jimmy Wu, Rika Antonova, Adam Kan, Marion Lepert, Andy Zeng, Shuran Song, Jeannette Bohg, Szymon Rusinkiewicz, and Thomas Funkhouser. 2023. Tidybot: Personalized robot assistance with large language models. *Autonomous Robots*, 47(8):1087–1102.

Lingling Xu, Haoran Xie, Si-Zhao Joe Qin, Xiaohui Tao, and Fu Lee Wang. 2023. Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment. *arXiv preprint arXiv:2312.12148*.

Xinnuo Xu, Ondřej Dušek, Verena Rieser, and Ioannis Konstas. 2021. Agggen: Ordering and aggregating while generating. *arXiv preprint arXiv:2106.05580*.

Wei Zhao, Goran Glavaš, Maxime Peyrard, Yang Gao, Robert West, and Steffen Eger. 2020. On the limitations of cross-lingual encoders as exposed by reference-free machine translation evaluation. *arXiv preprint arXiv:2005.01196*.

Author Index