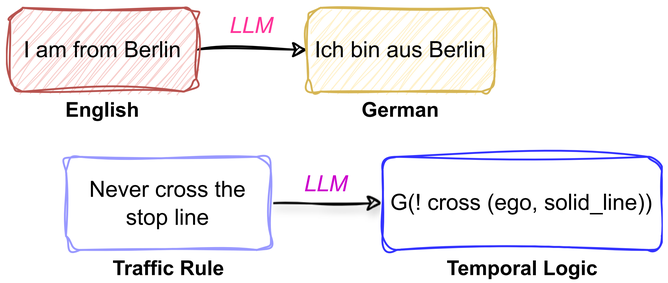
TR2MTL: LLM based framework for Metric Temporal Logic Formalization of Traffic Rules

Kumar Manas [1,2] Stefan Zwicklbauer [2] Adrian Paschke [1]

[1]Free University Berlin, Germany [2]Continental Automotive Technologies GmbH, Germany

Poster: **MoPo1I6.7**

Motivation



- Creation of unambiguous machine-readable driving rules
- Natural language are open for interpretation, vague and context-dependent
- Rule representation choice: Temporal Logic
- Automated rule representation using large language models (**LLMs**)

Problem and Solution

- Unavailability of formal traffic rule datasets
- Handcrafted formalization are not scalable

Solution:

1. Formalize traffic rules using Metric Temporal Logic (MTL), a logic system that can capture the temporal aspects of these rules, enabling precise and unambiguous interpretation.
2. Create a data-efficient formalization in the absence of a large dataset.

Handcrafted Traffic Rule Dataset and MTL

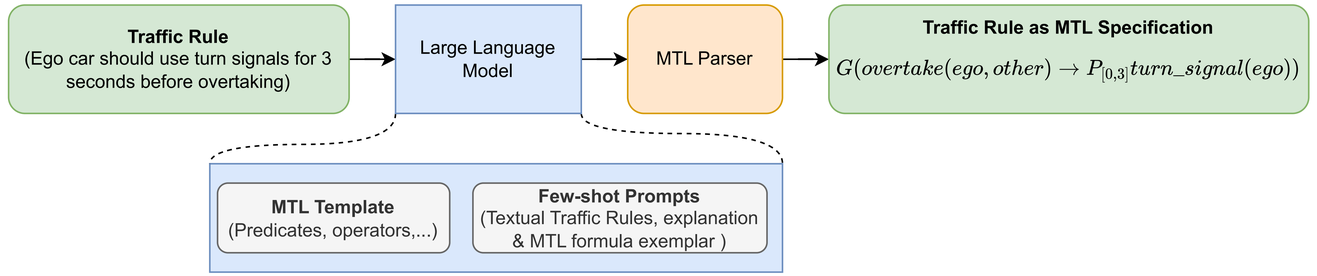
We created a dataset of handcrafted natural language and temporal logic pairs.

- **Source:** StVO [1] and UN Vienna traffic convention [2]
- **Dataset usage:** Evaluation and prompting of LLM-based automated formalization
- 50 pairs were created

Metric Temporal Logic (MTL)

- MTL allows specification of temporal constraints with precise timing intervals.
- MTL enables the accurate representation of real-world traffic scenarios, which often require specific timing.
- MTL is decidable for finite timed trace length.

Architecture Diagram



Methodology: Traffic Rule to MTL

- **Used MTL Operators:** "next (X)", "always (G)", "Until (U)", "at least once in past (O)", "eventually (F)", "past event relative to present (P)".
- The time interval for MTL operators is specified as a subscript, e.g., $F_{[t_1,t_2]}\varphi$, where φ represents atomic propositions and $[t_1, t_2]$ is time interval.
- **Prompting:** Chain-of-thought (CoT) designed for traffic rules as shown below

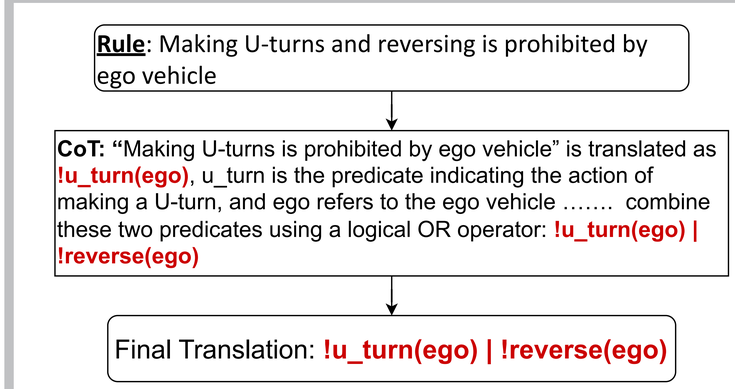


Fig: CoT Prompt illustration for Traffic Rules

1. Two-shot prompts examples are used in prompt file.
2. Prompt file also contains formal logic specific template, containing grammar of MTL.
3. Based on prompt file and MTL template LLM generates formalized traffic rules with explanation.
4. Finally LLM generated output is parsed using parser to generate more consistent and reliable representation of MTL before using them as monitors or other downstream task.

Results

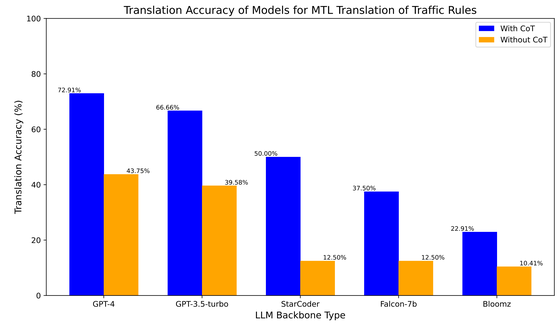


Fig: TR2MTL accuracy with and without CoT across various LLM backbone for traffic rule translation.

References

[1] StVO 2013 - nichtamtliches Inhaltsverzeichnis — gesetze-im-internet.de. https://www.gesetze-im-internet.de/stvo_2013/.

[2] unece.org. https://unece.org/DAM/trans/conventn/crt1968e.pdf.

kumar.manas@fu-berlin.de