Developing a Comprehensive Framework for Extracting and Formalizing Traffic Rules from Case Law

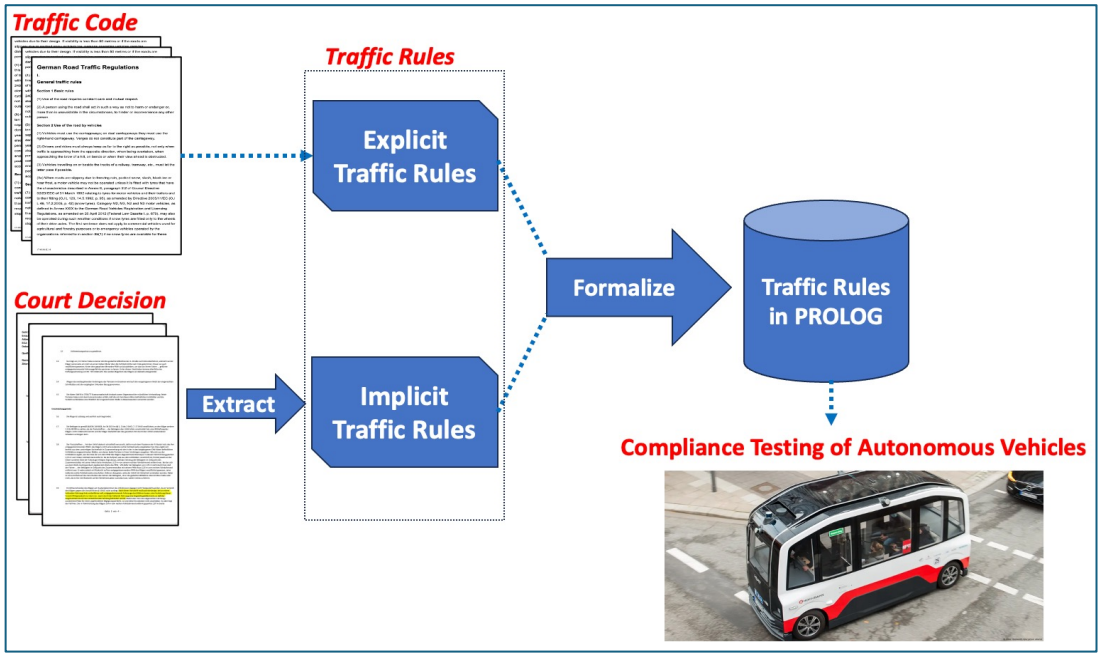
May Myo Zin

Center for Juris-Informatics, ROIS-DS

Introduction:

Laws are not static; they evolve over time. While laws can remain the same for extended periods, they are often subject to change through legislative processes, court rulings, or shifts in societal values.

This research focuses on traffic laws as an example of how legal systems must adapt to new challenges and advancements.



The rise of automated driving systems has the potential to revolutionize transportation, but their development faces the challenge of complying with a comprehensive and evolving set of traffic rules. This study aims to automate the extraction of implicit traffic rules from case law, integrate them into existing legal frameworks, and convert these rules into a format that autonomous vehicles can interpret.

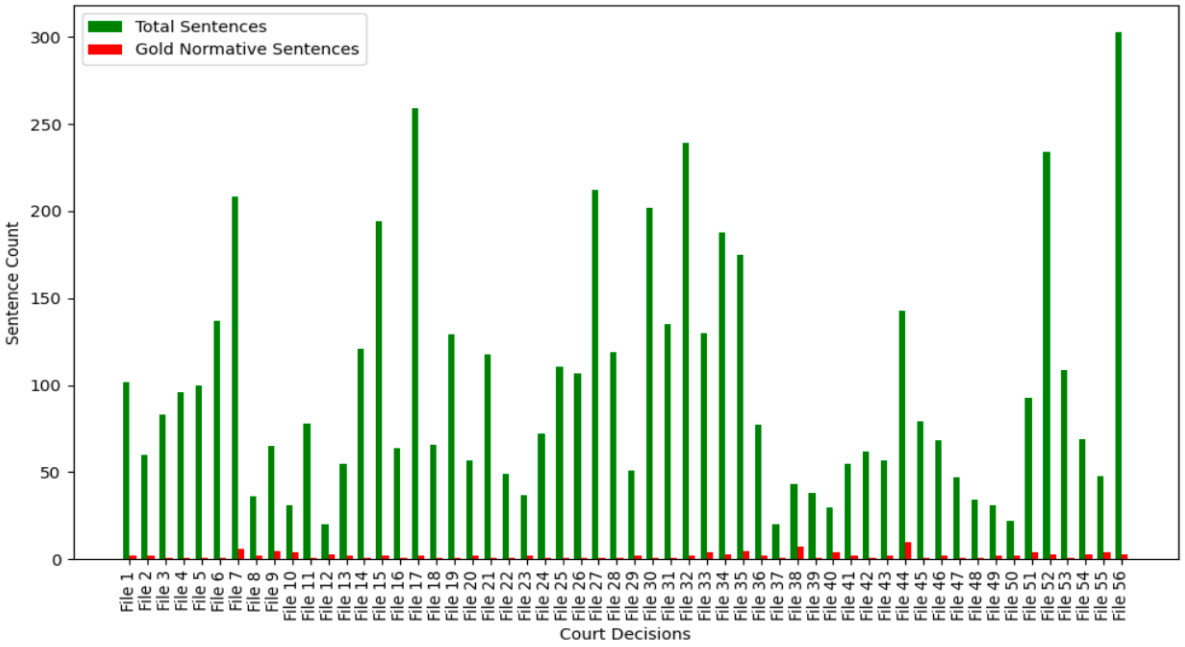


Figure 1. Total sentences vs. gold normative sentences per court decision.

Although a judgment may contain many sentences, typically only one or two sentences qualify as normative statements, highlighting the importance of efficient filtering and extraction techniques.

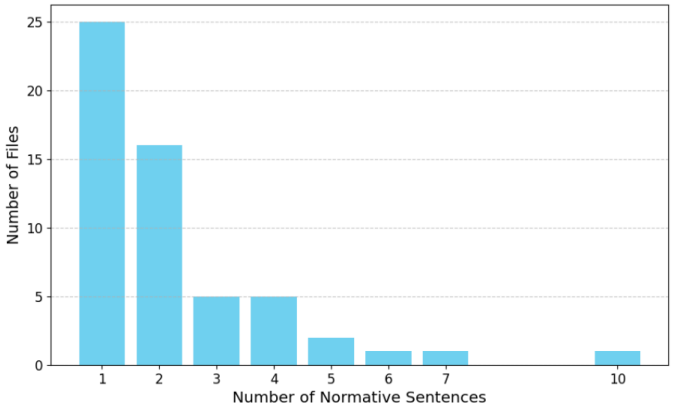


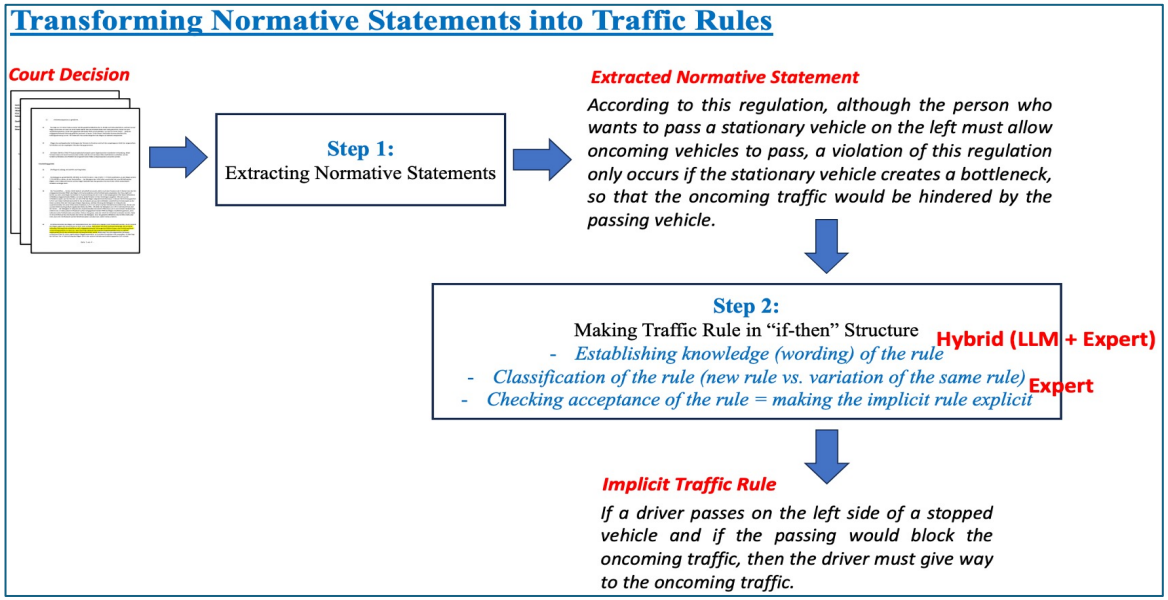
Figure 2. Number of files for each normative sentence count.

Results and Discussion (Normative Sentence Extraction)

Table 1. Performance comparison of four Prompt Engineering techniques in a zero-shot setting.

System	Precision	Recall	F2
Standard Prompt	0.177	0.792	0.467
Chain-of-Thoughts (CoT)	0.182	0.816	0.481
Layer-of-Thoughts (Lot)	**0.196**	0.824	0.501
Chain-of-Instructions (CoI)	0.193	**0.864**	**0.509**

- The evaluation dataset consists of 56 German court decisions, totaling ***6,242 sentences***, with only ***125*** identified as normative and specifically related to § 6 StVO.
- **With the top-k parameter set to 10, each approach extracts 560 sentences, of which at most 125 can be correct. Given this, Precision (P) is likely to be low ($P \leq 0.22$), but improving recall would still be beneficial for the system.**
  - Streamlining the Review Process: 11x Efficiency Boost
  - Cutting Workload by 91%
- This task also provides an opportunity to explore additional classification methods for filtering out irrelevant sentences from the final extracted (top-10) list, further enhancing the Precision score.



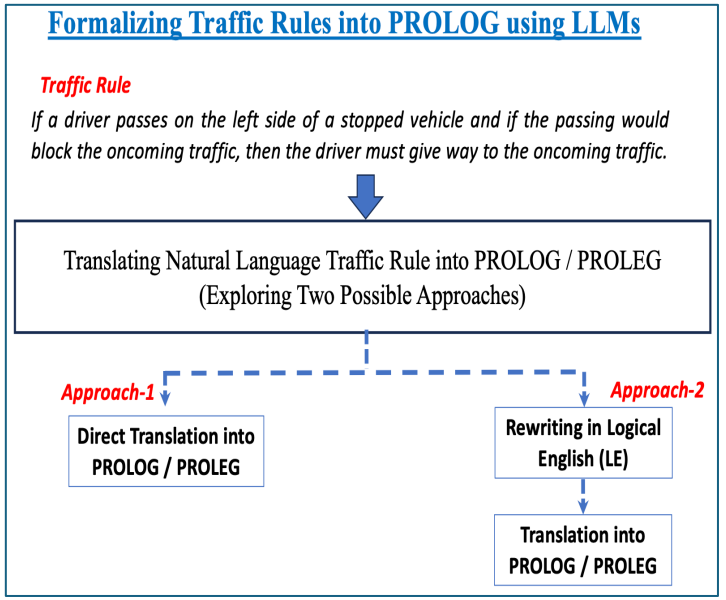


Table 2. Examples of Traffic Rules with their Corresponding Logical English Statements and PROLOG Representations

Narural Language Traffic Rule	Traffic Rule in Logical English	PROLOG Representation
If a driver is driving on a lane with a solid white line and the lane is so narrow that overtaking is not possible without crossing the solid white line, then he must not initiate an overtaking manoeuvre in the first place.	It is prohibited that a driver initiates an overtaking manouevre if the driver drives on a lane and a left lane is separated from the lane by a line that is solid white and it is not possible that the driver overtakes a vehicle unless the driver crosses the line because the lane is narrow.	prohibited(initiate(Driver,overtaking_manouevre)):- driving_on(Driver,Lane), separated_by_line(LeftLane, Line, Lane), solid_white(Line), causes(narrow(Lane), not_possible(and(overtake(Driver, Vehicle), not(cross(Driver, Line))))).
If a driver is driving on a carriageway with a solid white line, then a properly stored cargo may protrude above the solid white line if otherwise traffic to the right of the vehicle would be endangered, but oncoming traffic would not.	It is permitted that an object protrudes above a line if the driver drives on a lane and a left lane is separated from the lane by a line that is solid white and the line is a centreline and the driver drives a vehicle and the object is cargo of the vehicle and the object is properly stored and there is any traffic on the right of the vehicle and the object protrudes above the line so that the vehicle avoids danger to the traffic and for all cases in which there is any other traffic on the left of the vehicle it is not the case that the object is dangerous to the other traffic because the object protrudes above the line.	permitted(protrude_above(Object,CentreLine)):- drives(Driver,Vehicle), separated_by_line(OncomingLane,CentreLine,Lane), solid_white(CentreLine), driving_on(Driver,Lane), oncoming(Lane, OncomingLane), properly_stored(Object, Vehicle), right_of(Vehicle, Traffic), do_by(avoid(danger_to(Traffic)), protrude_above(Object,CentreLine)), forall(left_of(OtherTraffic, Vehicle), not(cause(protrude_above(Object,CentreLine), danger_to(OtherTraffic))))).

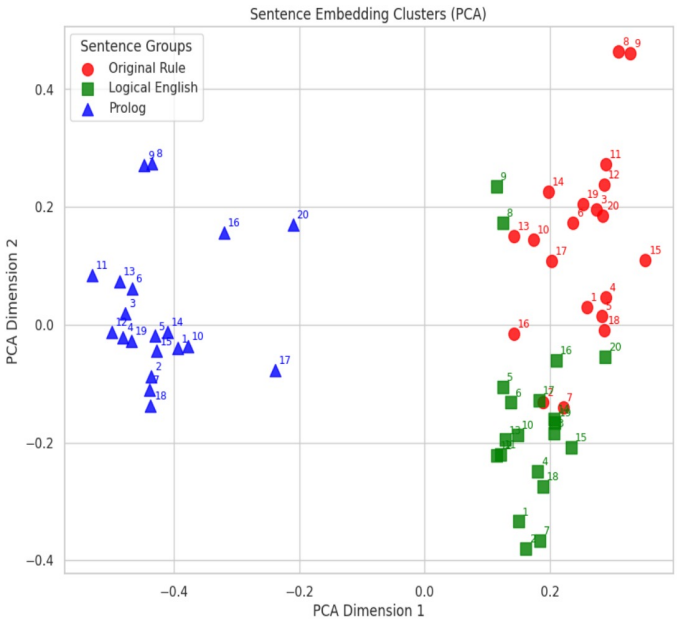


Figure 3. Semantic Gap between Original Traffic Rules, Logical English (LE) Versions, and PROLOG Representations measured using Sentence Embeddings.

Traffic Rule Formalization and Data Augmentation

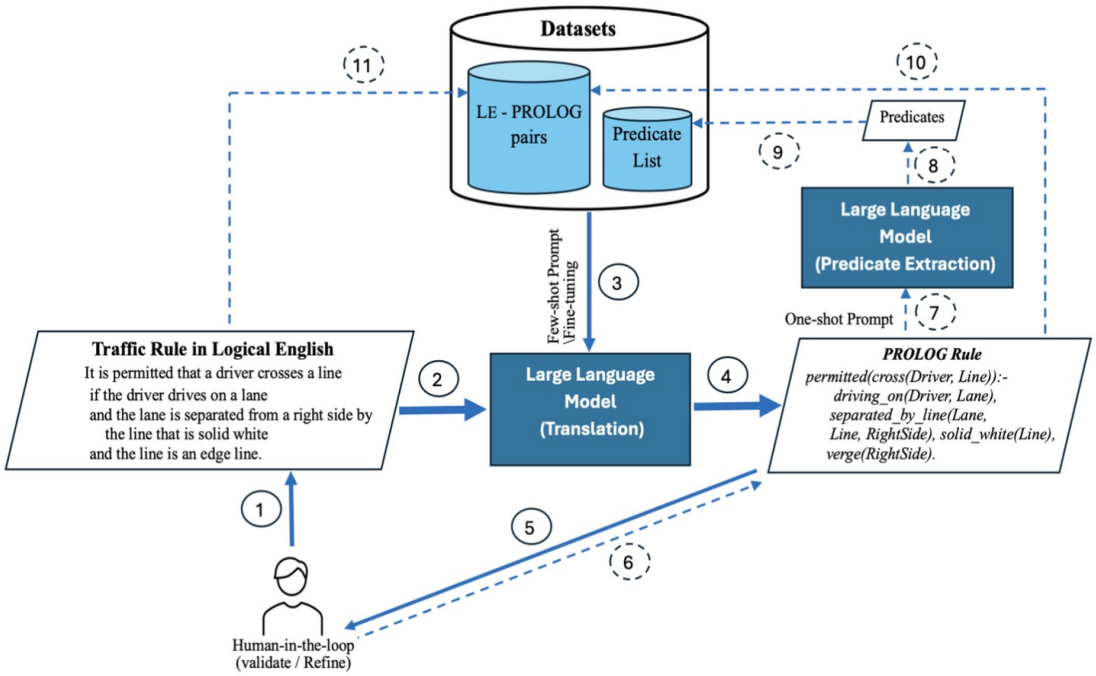


Figure 4. Human-in-the-Loop Pipeline Showing Translation Process (Steps 1–5, Solid Lines) and Training Data Expansion (Steps 6–11, Dashed Lines).

Experimental Result (Formalizing Traffic Rule into PROLOG)

Table 3. Comparative Accuracy of Advanced LLMs in Translating Logical English to PROLOG

Approach	GPT-4o	Gemini	Llama
Few-shot	0.75	0.75	0.70
Few-shot + Pre-defined predicates	0.85	0.80	0.75
Fine-tunning + Zero-shot	0.80	-	-

Conclusion

- Although the current approach does not achieve 100% accuracy in extracting rules from court decisions and translating them into Logical English or PROLOG, especially in low-resource settings, it significantly reduces the need for manual effort.
- Human intervention is only required for verification and minor adjustments, making the process more efficient and scalable over time.
- As the system continues to improve, the burden of manual labor will further decrease, streamlining legal text analysis and rule translation.