Published: 13 January 2026

Citation in BibTeX format

ICAIL 2025: 20th International Conference on Artificial Intelligence and Law
June 16 - 20, 2025
IL, Chicago, USA

RESEARCH-ARTICLE

You Take the High Road, and I'll Take the Low Road: Large Language Models Logical English and the Highway Code

THIAGO DAL PONT, University of Bologna, Bologna, BO, Italy

GALILEO SARTOR, Swansea University, Swansea, West Glamorgan, Wales, U.K.

ADAM WYNER, Swansea University, Swansea, West Glamorgan, Wales, U.K.

GIOVANNI SARTOR, University of Bologna, Bologna, BO, Italy

Open Access Support provided by:

University of Bologna

Swansea University

https://doi.org/10.1145/3769126.3769254
ISBN: 9798400719394

You Take the High Road, and I'll Take the Low Road

Large Language Models Logical English and the Highway Code

Thiago Dal Pont
Alma Mater Research Institute for Human-Centered Artificial Intelligence
University of Bologna
Bologna, Italy
thiago.raulino@unibo.it

Galileo Sartor
Swansea University
Swansea, United Kingdom
galileo.sartor@swansea.ac.uk

Adam Wyner
Swansea University
Swansea, United Kingdom
a.z.wyner@swansea.ac.uk

Giovanni Sartor
Department of Legal Studies
University of Bologna
Bologna, Italy
giovanni.sartor@unibo.it

Abstract

Autonomous vehicles (AVs) must comply with regulatory frameworks to ensure road safety and predictability. Current proposed AV systems predominantly rely on machine learning models that lack explicit, computable representations of traffic laws, raising concerns about accountability and robustness in complex scenarios. This study proposes a novel pipeline that embeds formal logic rules in the autonomous agent to ensure legal compliance. To address the knowledge acquisition bottleneck, we propose using Large Language Models (LLMs), robust prompt engineering, and Logical English (LE) to translate traffic rules from natural language into a human-readable, executable rule-based framework. The pipeline includes an error correction phase to refine the process of extracting legal rules, which are then integrated into a simulation environment. Our approach successfully performed the translation of legal text into a structured, computable format, improving the transparency and interpretability of the high level decision making. The error correction phase improves rule accuracy, while simulations further validate rule compliance and performance in dynamic traffic scenarios.

CCS Concepts

• **Theory of computation** → **Constraint and logic programming**; **Automated reasoning**; **Logic and verification**; • **General and reference** → **Evaluation**; • **Information systems** → **Language models**.

Keywords

LLM, Prompt Engineering, Natural Language Processing, Logical Representation, Legal Rules, Autonomous Vehicles

ICAIL 2025, Chicago, IL, USA

ACM ISBN 979-8-4007-1939-4/25/06
https://doi.org/10.1145/3769126.3769254

ACM Reference Format:
Thiago Dal Pont, Galileo Sartor, Adam Wyner, and Giovanni Sartor. 2025. You Take the High Road, and I'll Take the Low Road: Large Language Models Logical English and the Highway Code. In *20th International Conference on Artificial Intelligence and Law (ICAIL 2025), June 16–20, 2025, Chicago, IL, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3769126.3769254

1 Introduction

Autonomous vehicle technologies (AVs) have been rapidly developing over the last several years. In some approaches, e.g., Ford's Blue Cruise, AVs drive on specialised, controlled roads and largely separated from human drivers. However, we conjecture over the near term AVs and human agent vehicles (HVs) will share many common driving environments. In these circumstances, AVs and HVs will have to interact with one another, much as HVs themselves currently do. Such interactions will need not only to be predictable and understandable, but also abide by related legal regimes; that is, AVs and HVs both need to comply with traffic laws [20]. This is a necessary requirement to ensure safety, limit risks, and correlate liability for all road users.

Currently, most AV implementations rely predominantly on machine learning (ML) models to govern vehicle behaviour and ensure legal compliance. These systems lack an explicit knowledge representation of legal rules, meaning that AVs do not possess a structured, rule-based understanding of traffic laws. Instead, compliance is derived through data-driven learning processes during training, where the vehicle generates control outputs based on learned patterns of driving behaviour. In this context, lawful driving can be induced by manipulating the loss function, i.e., assigning penalties to harmful or unlawful behaviour. Most of the existing decision-making systems for AVs only consider collision-free driving as a safety condition or simply add some parameter settings that consider traffic laws Ma et al. [16]. The important role of deep reinforcement learning in the motion planning of autonomous vehicles has been addressed by Atakishiyev et al. [2]. While effective in dealing with many practical situations, this approach raises concerns regarding explainability and robustness in complex factual, legal, and ethical scenarios. It is also unclear how effective this approach is in very complex environments with different actors all sharing the road and where the data is "thin".

While machine learning approaches have a role, we claim that AVs should also have a computable representation of traffic rules, satisfying the following three requirements:

- they should capture the content of traffic laws directed to humans
- they should be automatically executable
- they should be understandable to humans

An explicit, computable representation of traffic rules is indeed essential for ensuring AVs' legal compliance, predictability, and interpretability. Unlike learning-based methods that derive compliance from statistical correlations in training data, a structured legal framework enables rule-based transparency for verification and regulatory oversight. Explicitly encoding traffic rules enhances AV-human interactions, by making AV decisions interpretable and aligned with human expectations. Moreover, having the executable rules correlate to humanly interpretable rules, we provide a common ground of understanding and action in the driving environment and in ascribing liability. By having such a legal common ground, AVs and HVs are, to a greater degree than otherwise, equal before the law [22].

This paper addresses the long-standing *knowledge acquisition bottleneck* [3]. In the current context, this is the problem of systematically translating the source legal text into some formal representation. Early efforts, e.g., the encoding of the British Nationality Act as a Logic Program [23], were manual and ad hoc. Efforts to make the extraction of information structured and rule-based were proposed by [30], yet only in a fragmentary way. In [29], a range of such approaches was outlined. Some proposed a normalised standard, e.g., Semantics of Business Vocabulary and Business Rules [24] and Simplified Technical English; the issue is whether these restricted languages are suitably flexible for traffic laws and in a flexible manner. Others used well-developed parsing with semantic representation, such as Attempto Controlled English [10] or C&C/Boxer [28]; but, such approaches faced the complexity of validating the output representations. The knowledge acquisition bottleneck remains a problem, particularly when scaling up, in realising formal representations of legal information.

The current paper aims to address the bottleneck by leveraging large language models (LLMs), prompt engineering, and a suitable target controlled natural language (CNL) *Logical English*[15], which like Attempto and C&C/Boxer, translates to Prolog so as to be executable. Like Ruleburst (Oracle Policy Automation) [8], which provides "just enough" representational power, Logical English constrains the variety of constructions to predicate-argument expressions and connectors. Unlike other systems, the predicate-argument structure can be developed "on the fly".

LLMs are models trained on large text datasets to understand and generate human-like language. Using transformer architectures, they have achieved state-of-the-art performance in various tasks, including translation, code generation, and text summarisation [31]. Their ability to extract structured information from unstructured text makes them particularly valuable in areas such as legal and regulatory compliance [4]. To mitigate the known issues with LLMs, namely hallucinations and biases, prompt engineering plays a crucial role. Various prompting techniques improve LLM performance by structuring input to promote accuracy [12, 27]. Key approaches include the following:

Prompt engineering enhances LLM performance by structuring input for accuracy [12, 27]. Standard prompts use direct question-answer pairs for simple tasks, while Chain of Thought (CoT) improves complex reasoning by breaking tasks into steps. Layer of Thought (LoT) refines responses through hierarchical constraints, enhancing accountability and multi-turn interactions. Chain of Instructions (CoI) provides explicit steps, reducing ambiguity and improving adherence to procedures.

The main novel contribution of the paper is the development of a pipeline to facilitate the translation of source legal language into expressions of a normalised natural language (Logical English), which can then be automatically transformed into an executable rule-language (Prolog). The pipeline uses LLMs (LLama 3.3 70B) and prompt engineering. An important new aspect is an error correcting phase, which identifies missing constructions that can be added as Logical English templates. The Logical English output is translated into Prolog, and passed to a simulation environment based on NetLogo[1], a multi-agent programmable modelling environment of roads and intersections that the vehicles move through. This simple simulation provides a visual representation of the behaviour and is also useful to generate metrics of the running system by keeping a log of the decisions made by the vehicles [2].

The scope of the paper is restricted to the translation of the source legal text into Logical English. Other relevant and crucial topics remain for future work, e.g., the treatment of norm violations.

Section 2 presents the methodology, covering the prompting approach, the dataset and the pipeline. Section 3 describes several experiments and their results, with a particular focus on the syntax checker. The evaluation, discussed in section 3.2, includes evaluations by researchers with varying levels of expertise in Logical English and LLMs. Section 4 compares this work with related research, and finally, Section 5 presents the conclusions and future works.

2 Methodology

In this section, we discuss our methodology, dataset, and pipeline.

2.1 Logical English

The CNL selected is Logical English (LE), a template-based language built on Prolog[15]. Logical English canbe regarded as syntactic sugar for Prolog, as it internally compiles to and runs the Prolog translation of the program. The use of Logical English makes the rules more accessible to those without a background in computing or logic, a goal that is very important in the context of autonomous vehicles, for the resulting system to be explainable in its high level decision making.

Logical English is a language based on templates, that express the accepted predicates and identifies their variable arguments (surrounded by asterisks). Variables are then defined in the context

[1]https://www.netlogoweb.org/

[2]The code for the different sections of the proposed pipeline (rules, simulators, prompts) are available at https://github.com/LyzardKing/mind-the-gap/tree/ICAIL2025

of rules by first introducing them with indefinite articles ("a", "an") followed by a common noun, and optionally an identifier name. The variables can then be recalled in the body of the rule by using "the", with the same common noun, or directly the identifier.

As a simple example, we can think of modelling the ancestor rule as follows:

Listing 1: Sample Logical English program

```
the templates are:
*a person* is a parent of *a person*.
*a person* is an ancestor of *a person*.

the knowledge base parent includes:
a person P is an ancestor of a person Q if
P is a parent of Q.

a person P is an ancestor of a person Q if
a person R is a parent of Q
and P is an ancestor of R.
```

The use of templates is somewhat rigid in ensuring representation as it makes the language arguably stricter than some others. While other templates may be added differently, we limited their number, instead attempting to simplify the natural language clauses.

The use of LE makes it possible to share the representation with humans and machines, with a common understanding of the rules [13], an important issue in the shared space in which these different agents will move. While there are multiple other CNLs, they each have different goals and characteristics. Some enable a more detailed grammar, enabling the expression of complex syntactical structures such as composite sentences and relative clauses[11], while others focus on shorter predicates, exchanging some verbosity with a simpler, more restricted, representation, useful to express concepts to users with different knowledge of English[32].

For the reasoning phase, Logical English automatically translates the CNL to Prolog. The output is then converted back to natural language and presented to the user. For further details on the CNL to Prolog translation, see [15].

2.2 Chain-of-Thought Prompting

Given the need for precision in parsing, we use a Chain-of-Thought (CoT) based prompting strategy to systematically extract, structure, and translate traffic rules into CNL. Our approach ensures that each step in the parsing process is explicitly defined, reducing ambiguity and improving consistency. The process begins by decomposing regulatory sentences into structured elements before mapping them to LE. We describe and detail the strategies. Sentences are first broken down into individual components such as subjects, conclusions, deontic modalities (e.g., must, can), and conditions. The extracted data is then formatted in JSON and mapped to LE templates. Finally, these structured components are recombined using logical operators (e.g., AND, OR) to reconstruct the legal rule in a formalised, machine-readable format. To assess the effectiveness of this approach, we experiment with prompts both with and without few-shot examples, analysing their impact on syntactic accuracy and semantic coherence.

The first prompt strategy takes a general approach to transforming legal text into LE. It provides the model with a conceptual overview of LE, including its structural elements, templates, and logical operators. The prompt guides the model in structuring sentences into logical rules, but does not explicitly define the intermediate steps of reasoning. Instead, it allows the model to determine autonomously how to derive the final structured output. This approach prioritises flexibility by ensuring that the extracted rules conform to the LE syntax and maintain logical consistency. However, because it does not impose explicit procedural constraints on the extraction of key components such as subjects, conditions or modalities, the model's output varied slightly from trial to trial. This inconsistency stemmed from the lack of a fixed reference, which led to fluctuations in the generated rules.

To address this issue, we introduced a second prompting strategy using 1-shot learning, in which a single example, rule 171 from the UK Highway Code, was manually constructed step by step. This refinement provided the model with a concrete reference, reducing output variability and ensuring greater consistency in the logical transformation process. This method systematically decomposes the input into distinct components. Sentences are segmented; for each segment the model extracts key elements, including subjects, conclusions, deontic modalities and conditions. These extracted elements are then formatted in JSON and mapped to pre-defined LE templates, ensuring consistency and accuracy in the output. In addition, few-shot examples are incorporated at each step to reinforce the expected output format.

By limiting model's *autonomy*, this approach tries to minimise ambiguity and hallucination while enforcing strict adherence to logical structuring and consistency in template selection. The prompt overview is presented in Listing 2.

Listing 2: Summary Step-by-Step Structured Parser Prompt

```
## Objective
Convert natural language text into Logical English (LE)
    through a structured, step-by-step extraction
    process.

## Approach

1. **Sentence Segmentation**: Split the input legal text
    into individual sentences.
2. **Structured Information Extraction**: For each
    sentence, explicitly extract:
    - Subject (the entity performing the action)
    - Conclusion (main action in the sentence)
    - Deontic Modality (must/should/can, if applicable)
    - Conditions (contextual requirements for the rule to
     apply)
3. **JSON Representation**: Present extracted elements in
     a structured JSON format.
4. **Template Assignment**:
    - Match extracted conclusions and conditions to
    predefined LE templates.
    - Only create a new template if no suitable
    predefined template exists.
5. **Rule Construction**: Use the assigned templates to
    build Logical English rules.
6. **Few-Shot Learning**: Provide explicit examples
    demonstrating each step to guide the model's output.

[Final Remarks on task to be performed, the steps, and
    their expected output.]
```

In addition to defining the system prompt to guide the model through the task, a structured input format is crucial to ensure proper processing. The input should clearly differentiate between the new sentence to be analysed and the existing set of templates, allowing the LLM to exploit prior structured knowledge, while retaining the flexibility to introduce new logical constructs when necessary. The proposed format is presented in Listing 3.

Listing 3: Input structure for parsing new natural language text.

```
## Sentence to Analyse
[Raw text to be parsed]

## Previous set of templates
[List of templates extracted beforehand.]
```

The target model for the prompts provided is LLaMA 3.3 70B, an open-weight LLM selected for its optimal balance of performance, computational efficiency, and accessibility. The use of open models promotes reproducibility, transparency and adaptability, allowing researchers to validate results and explore different configurations without the constraints of proprietary restrictions. LLaMA 3.3 70B ranks among the top 20 models in the General-Purpose Question Answering (GPQA) and MMLU benchmarks, as reported by Stats [25], demonstrating its proficiency in text processing. Model inference was primarily performed in a cloud-based configuration via HuggingChat[3], ensuring fast prototyping and wide accessibility for research.

For model parameters, we set the temperature to 0.6 to balance creativity and determinism, with higher values increasing variability. A Top-P value of 1.0 was used to define the probability distribution of word choices, while a Top-K of 50 restricted word choices to the most likely tokens, improving predictability.

2.3 Dataset

The dataset analysed in this study is a subset of the United Kingdom's Highway Code (HC).[4] Published by the UK Department for Transport, the HC is a reference for road users, including drivers, cyclists, and pedestrians, to promote safety on the road, whilst also supporting a healthy, sustainable and efficient transport system. All road users are presumed to be familiar with the contents of the HC. Its structured guidelines and simpler language makes it suitable for analysing legal rules and their logical formalisations for computational applications.

This study focuses on a specific subset of the UK HC which details the rules related to junctions (or intersections) as contained in the *Using the Road* section of the HC. These rules regulate the behaviour of road users, mainly vehicles, (with references to others, such as cyclists and pedestrians), ensuring safety and fluidity in complex traffic scenarios.

The data is scoped in several ways. While the HC addresses a wide range of situations and road users, this research narrows its scope to vehicular compliance at intersections. The behaviour of non-vehicular road users, such as pedestrians and cyclists, is outside the scope of this study. Certain rules are accompanied by additional legal references, mainly to the *Road Traffic Act 1988*, and *The Traffic Signs Regulations and General Directions 2016*. This is the case with rules that prescribe an obligation or prohibition, and the referred articles usually mention the penalty issued for breaking such a rule. However, such references and their content are outside the scope of the dataset considered here. Illustrative examples and diagrams in the text enhance clarity and practical applicability, but are not relevant to our analysis.

The UK HC has a consistent structure organisation across its different sections. Each rule is explicitly numbered and often accompanied by legal mandates ("must/must not"), advisory statements ("should / should not"). This format allows for parsing and facilitates the extraction of rules for computational processing.

In terms of linguistic features, the HC combines precision with complexity, using highly specific terminology and syntactically complex constructions. Sentences often contain conditional clauses, lists, and embedded directives reflecting the nuanced requirements of traffic regulation. The text's reliance on modal verbs such as 'must' and 'should' presents the duality between the HC as a legal framework and as a practical guide for road users.

The initial dataset is comprised of about 12 rules, from the "Junctions" section of the HC and range from 30-70 words in length. In general, if a rule is longer, it is separated into paragraphs (or a list) in the original text. In this case, each portion is treated as a different rule. For example, rule 170 states "Take extra care at junctions. You should.", followed by a list of 7 recommendations for the driver of the vehicle approaching the junction. Most rules that describe an action (obligation or permission) are shorter, with emphasis on the deontic term, which is capitalised and in bold weight, as in rule 171, shown below.

> You **MUST** stop behind the line at a junction with a 'Stop' sign and a solid white line across the road. Wait for a safe gap in the traffic before you move off.

The structure of these rules can be mapped to a set of components, which are similar to the model of [30]:

- Subject: Generally the term "you";
- A deontic modality: terms such as "must", "can". This term might not be present, and if it appears it is often in bold typeface;
- The action: the action that is enforced, denied, or permitted, with additional specifications (in the example above "stop behind the white line";
- The conditions under which the action is performed (the presence of a 'Stop' sign and solid white line).

This analysis of the source text has two main impacts:

- The extracted rules have a stricter design, so they are modelled in a way that is consistent;
- The LLM can be instructed in a more precise way with regards to the identification and extraction of rules, keeping track of the reasoning

The model according to which we design the rules would thus make use of the deontic modal expression ("must" in the example) as the head of the rule, with the subject, action and properties following as arguments. The same structure can be modelled in Logical English, and its use would enable a more compact list of templates. Additionally, we are keeping the point of view (as in the original text) on the agent deciding to act. What is called in the literature the "ego" vehicle, we substitute "ego" for "you" in

[3]Available at https://huggingface.co/chat/. It facilitates using a range of open-weight LLMs while allowing tweaking with the main hyperparameters, and creating what is termed *assistant*, that is, LLMs with specific system prompts and purposes.
[4]https://www.gov.uk/browse/driving/highway-code-road-safety. Accessed 2 May 2025.

Listing 4: Obligation and Permission in Logical English

```
ego must stop behind the line at a junction
if the junction has Stop sign
and the junction has solid white line across the road.

ego can move off
if safe gap exists in traffic.
```

the original text "You **MUST** stop...." and add "ego" for the missing subject of an imperative "Wait...."

As an example, the previously presented rule would be modelled as in Listing 4.

As we can see in the difference between Rule 170 and 171, the HC contains different types of rules, and the impact that those rules would have on AVs may differ. In particular, we have rules that explicitly state a desired behaviour (Rule 171), and also rules that give indications that are meant to inform the driver of how they should use or prioritise their sensing ability.

As we can see, rule 170 instructs drivers that certain road users are considered more vulnerable, and thus the driver should pay extra attention. Additionally, it states that the driver should not make assumptions on how the other road users (cyclists and pedestrians) will move.

While portions of Rule 170 can be modelled and transformed in logic rules, it is yet unclear how more general rules should apply. For example, if we assume that the AV is constantly checking its entire surroundings, what does an instruction like the following add to its behaviour:

> [...] watch out for cyclists, motorcyclists and pedestrians including powered wheelchairs/mobility scooter users as they are not always easy to see

While portions of Rule 170 can be modelled and transformed into logic rules, it is yet unclear how such more general rules on perception and knowledge should be analysed. There is a certain degree of interpretation - we have to decide which rules to translate directly and which could instead be incorporated in the knowledge base as contextual information. As we will see in further examples, while such complex and as yet unresolved topics are observed, they are left for future work. Commonsense reasoning is important in a context such as driving, where even humans rely on tacit knowledge.

2.4 Pipeline

The pipeline starts with one or both of the following inputs: (a) a raw article from the UK HC (as described in section 2.3); or (b) a pre-existing set of templates, where the system evaluates new articles in relation to those already extracted, thereby preventing the generation of multiple Logical English structures for the same situation or expression. Each article describes driving behaviour and associated expectations. When using an existing template set, the system evaluates how the new rule interacts with previously validated rules.

Once the input is structured, an LLM parses the source text into LE following a structured prompt (see section 2.2). However, generating a rule in LE is not enough. The output must first undergo *syntax validation*, which works in a similar way to a compiler analysing programming languages. This phase ensures grammatical and logical correctness by enforcing compliance with predefined syntactic constraints. If inconsistencies are detected, the system generates detailed feedback, including the original input, the generated LE expression, and a description of the errors. The LLM then attempts to correct these problems, iterating until the rule meets all structural requirements.

After successfully passing syntax validation, the rule is tested against atomic driving scenarios to verify its expected behaviour. These scenarios are sets of facts with an expected outcome. If a rule produces unexpected results or fails to generalise appropriately (that is, there is no correspondence between the input facts in the scenario and the conditions of the rules), a scenario error is triggered. This initiates a feedback loop in which the LLM refines the rule based on the error descriptions, iterating until all tested scenarios match the expected behaviour. These scenarios can be tailored to test issues in the representation of norms, as well as the expectation of the evaluators. The same rule can have different failure states, so the more scenarios we can use to test generated rules, the more we can expect them to be in line with the original norm.

Once the rules have passed atomic scenario testing, they are integrated into a traffic simulation environment with multiple agents and actions such as NetLogo or SUMO (Simulation of Urban Mobility)[5]. While prior validation ensures that a rule works correctly in isolation, multi-agent simulation evaluates how multiple rules interact in more dynamic, real-world conditions involving drivers, pedestrians and cyclists. Unlike earlier stages that focus primarily on correctness, simulation provides insight into potential unintended consequences, inconsistencies, or inefficiencies that require further refinement.

Following the computational and scenario-based evaluations, the rule undergoes an expert review, where legal and transport specialists assess its compliance with legal requirements, its impact on road safety, and the mitigation of unintended consequences. Only after passing this final review is the rule added to the evolving dataset of validated traffic rules. This analysis is out of scope in this paper.

This iterative pipeline, combining LLM capabilities, rigorous validation, simulation-driven insights and expert evaluation, ensures that each rule generated is not only formally correct, but also practically effective in real-world applications.

3 Experiments

In order to evaluate our proposed pipeline, we carried out two *qualitative* analyses. The first analysis focuses on a case study using a specific rule (Rule 175) from the UK Highway Code. By applying the pipeline step by step, we illustrate its operation and evaluate the results, in particular at the LLM conversion and syntax checking stages. Due to space and time constraints, the steps beyond syntax checking will be considered in future work. This analysis provides a clear demonstration of how the pipeline works in practice.

In the second analysis, we present an overview of human evaluation. By combining these two analyses, we aim to provide a

[5] https://eclipse.dev/sumo/

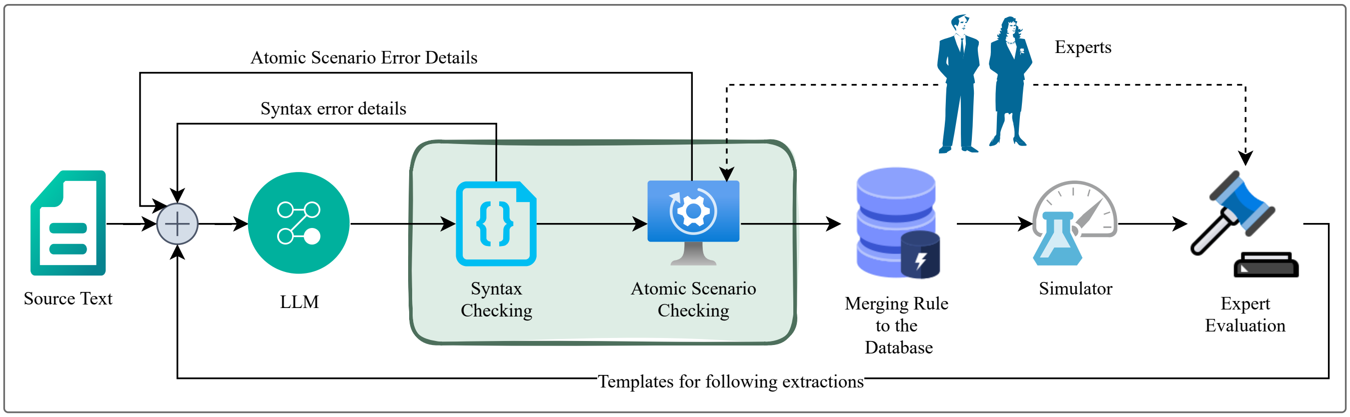

Figure 1: The Processing Pipeline

comprehensive assessment of the performance of our system, while maintaining clarity for the reader.

3.1 Case Study of Rule 175

In this section, we present a detailed example (self-contained and isolated from the other experiments) of a complete LLM output, demonstrating its step-by-step reasoning based on the given instructions until it produces the final LE output.

We focus on Rule 175 of the UK Highway Code, reproduced below:

> You MUST stop behind the white 'Stop' line across your side of the road unless the light is green. If the amber light appears you may go on only if you have already crossed the stop line or are so close to it that to stop might cause a collision.

The LLM is provided solely with this rule as input and the system prompt. As illustrated in Listing 5, the complete output of the model is presented. The response adheres to Markdown syntax, a formatting system that ensures proper rendering in a typical browser interface.

Listing 5: LLM Generated output for Rule 175 of the UK HC

```
## Step 1: Break the input into sentences.

The input can be broken down into two main sentences:
1. You MUST stop behind the white 'Stop' line across your
     side of the road unless the light is green.
2. If the amber light appears you may go on only if you
     have already crossed the stop line or are so close
     to it that to stop might cause a collision.


## Step 2: Extract relevant information for each sentence
     .

### Sentence 1:

- Subject: "you"
- Conclusion: "stop behind the white 'Stop' line"
- Deontic Modality: "must"
- Conditions:
  * "the light is not green"


### Sentence 2:

- Subject: "you"
- Conclusion: "go on"
- Deontic Modality: "may"
- Conditions:
  * "the amber light appears"
  * "you have already crossed the stop line"
  * "you are so close to the stop line that stopping
     might cause a collision"

## Step 3: Present the JSON with the extracted elements
     for each sentence.

```json
[
  {
    "subject": "ego",
    "conclusion": "stop behind the white 'Stop' line",
    "deontic_modality": "must",
    "conditions": [
      "the light is not green"
    ]
  },
  {
    "subject": "ego",
    "conclusion": "go on",
    "deontic_modality": "may",
    "conditions": [
      "the amber light appears",
      "you have already crossed the stop line",
      "you are so close to the stop line that stopping
    might cause a collision"
    ]
  }
]
```

Step 4: Identify the templates for each JSON object.

Sentence 1:

Conclusion: "you must stop behind the white 'Stop'
     line" - `*an agent* must *an action* at *a location*
     `
Conditions:
- "the light is not green": `*a thing* is not *a property
     *`

Sentence 2:

Conclusion: "you may go on" - `*an agent* can *an
     action*`
Conditions:
```

```
- "the amber light appears": `*an event* occurs`
- "you have already crossed the stop line": `*an agent*
    has *a property*`
- "you are so close to the stop line that stopping might
    cause a collision": `*an agent* is in *a state*`

## Step 5: Build the rules using the identified templates
   .

1. You MUST stop behind the white 'Stop' line across your
    side of the road unless the light is green.

```le
the knowledge base includes:
ego must stop behind the white 'Stop' line
if the light is not green.
```

2. If the amber light appears you may go on only if you
    have already crossed the stop line or are so close
    to it that to stop might cause a collision.

```le
the knowledge base includes:
ego can go on
if the amber light appears
and (ego has crossed the stop line
or ego is so close to the stop line that stopping might
    cause a collision).
```

The final answer is:

```le
the knowledge base includes:
ego must stop behind the white 'Stop' line
if the light is not green.

ego can go on
if the amber light appears
and (ego has crossed the stop line
or ego is so close to the stop line that stopping might
    cause a collision).
```
```

Notice that the output generally follows the set of pre-defined guidelines: a) breaking the input into sentences; b) for each sentence, extracting the relevant information (subject, conclusion, deontic modality, conditions); c) presenting the structured JSON with the extracted information; d) assigning templates for each JSON object, e) building the rules and present the final LE structure.

In this initial translation, the LLM is not always able to identify the correct template to use or makes some simple syntax errors in the same templates. As an example, following the same rule as before, the initial output of the system was as follows:

Listing 6: Incorrect LLM-translated Rule 175 in LE

```
the templates are:
*a thing* has *a property*.
*an agent* must *an action* at *a location*.

the knowledge base example includes:
ego must give way to traffic on the main road
if the junction has broken white lines across the road.
```

While the content is correct, the LLM did not correctly use the generated template, using "on" the main road instead of "at". Though relatively minor, this is captured and reported as a syntax error when loading the generated rule-base. The error can then be sent to the LLM with the original text, as shown in Listing 7.

The LLM will return a new output, with the step-by-step reasoning, a suggested output, and the reason for the change (See Listing 8). The code passes through the syntax checking, and if it goes to atomic scenario checking. As can be seen in the example, by passing to the LLM the error message containing the line where the issue occurred, as well as the original text and the incorrect LE representation, the system is able to identify that the issue is that the predicate does not fit the template, and correct it, by either suggesting a new template or modifying the code. Where there are different possibilities the system can also provide alternative implementations, depending on the level of detail needed in the representation.

Listing 7: Input prompt employed during the syntax correction phase of Rule 175 translation.

```
# Syntax error checking

Fix the syntax error. Be sure that the rules follow the
    templates and the general rules presented in the
    prompt.

## Sentence to analyse
The approach to a junction may have a 'Give Way' sign or
    a triangle marked on the road. You MUST give way to
    traffic on the main road when emerging from a
    junction with broken white lines across the road.

## Generated LE
the templates are:
  *a thing* has *a property*.
  *an agent* must *an action* at *a location*.

the knowledge base example includes:
ego must give way to traffic on the main road
if the junction has broken white lines across the road.


## Error generated
ERROR: _6592:9: Syntax error: LE error found around this
    expression: : ego must give way to traffic on the
    main road
```

Listing 8: Output from the error correcting LLM phase for Rule 175

```
## Step-by-step analysis of the problem:

- Identify the error location: The error message points
    to the expression "ego must give way to traffic on
    the main road". This suggests that the issue lies in
     the formulation of this sentence according to the
    Logical English (LE) syntax.
- Review LE templates: The provided templates are:
    - a thing has a property.
    - an agent must an action at a location.
- Analyze the problematic sentence: The sentence "ego
    must give way to traffic on the main road" is
    supposed to fit the template "an agent must an
    action at a location". However, "give way to traffic
     on the main road" does not neatly fit into the "an
    action" slot because it implies an action ("give way
    ") directed towards another entity ("traffic"),
    which is not directly accommodated by the simple "an
     action" template.
- Adjust the sentence to fit the template: To align with
    the template, we need to ensure that the action is
    clearly defined and that any additional details are
    properly integrated. The action here is "give way",
    and it's directed towards "traffic on the main road
    ". This suggests we might need a more nuanced
    approach to represent the action and its object.
```

```
## Fixed solution:

the templates are:
*a thing* has *a property*.
*an agent* must *an action* to *an object* at *a location
     *.

the knowledge base example includes:
ego must give way to traffic at the main road
if the junction has broken white lines across the road.

## Explanation of changes:

Modified template: Introduced a new template "an agent
     must an action to an object at a location" to better
      fit the structure of the sentence, allowing for an
     action directed towards an object at a specific
     location.
Adjusted sentence: Changed "give way to traffic on the
     main road" to "give way to traffic at the main road"
      to fit the new template, making "traffic" the
     object of the action "give way" and "the main road"
     the location.
```

3.2 Human and Scenario-Based Assessments

To evaluate the performance of our pipeline, we conducted both human and scenario-based assessments. The logical consistency, syntactic accuracy, and fidelity of the translations were assessed by human reviewers, while rule performance in simulated environments was tested by means of scenario-based evaluation. The following sections delineate the methodology, evaluation criteria, and key findings from both approaches.

3.2.1 Human Assessment of Rule Translation and Correction. The goal of this assessment is twofold: (1) to evaluate how well the model extracts CNL rules following the methodology presented in previous sections, and (2) to evaluate its ability to correct errors after receiving feedback from the CNL compiler.

For this assessment, we selected four rules from the UK HC: Rule 172, Rule 173, Rule 175, and Rule 176. Three reviewers participated in the evaluation. All had experience with Logic Programming, but none were specialists in Logical English, but had some degree of familiarity with the CNL. This ensured a more general and unbiased assessment. Each reviewer independently evaluated the source text and CNL translations of the four rules, allowing for a more reliable and trustworthy analysis of the system's performance.

During the evaluation, the reviewers were provided with: (1) The original source text; (2) The LLM-generated CNL output; (3) A questionnaire to assess the accuracy and faithfulness of the rules.

Each question in the survey was answered in binary format: 1 (Yes) in the event that the requirement was met, and 0 (No) in the event that the requirement was not met. The survey addressed the following questions:

Listing 9: Human evaluation Survey

```
Q1. Is the step-by-step reasoning correct?
Q2. Are there any hallucinations that compromise the
     results?
Q3. Did the LLM correctly identify the error?
Q4. Did the LLM correctly fix the error?
Q5. Is the LE translation (after syntax checking) correct
     ?
Q6. Is the LE translation faithful to the source text?
Q7. Does the LE translation contain any information not
     in the source text?
Q8. Does the LE translation omit any relevant information
      from the source text?
```

Each question is designed to assess a specific evaluation dimension. $Q1$ addressed whether the LLM followed a coherent, step-by-step reasoning process without logical leaps or contradictions. $Q2$ relates to the presence of hallucinations, such as content fabricated by the model that has the potential to compromise the reliability of the output. $Q3$ and $Q4$ address whether the LLM correctly identified and addressed syntax errors that were flagged by the Logical English compiler. $Q5$ posits an evaluation of whether the final output adheres to the appropriate logical English syntax. $Q6$ concerns semantic fidelity, i.e. the degree to which the translation preserves the intent and meaning of the source rule. $Q7$ is concerned with determining whether the translation incorporates information not present in the original text, even in cases where it is syntactically valid, while $Q8$ addresses the opposite on whether important information from the source text was ignored by the LLM.

Following the collection of all reviewer responses, the results were compiled into Table 1, which presents the compilation of evaluation results. The three reviewers' evaluation is merged into one. In case of disagreement among them, the majority opinion was adopted and presented.[6] In instances where the model successfully generated syntactically valid output on the initial attempt, rendering correction unnecessary, the question was designated as "not applicable".

Table 1: Results from the human evaluation. For each rule-question pair, the majority evaluation is presented. X means Not Applicable

Question	TR 172	TR 173	TR 175	TR 176	Correct/Total
Q1	1	1	1	1	4/4
Q2	0	0	0	0	0/4
Q3	X	X	0	1	1/2
Q4	X	X	0	0	0/2
Q5	1	0	0	0	1/4
Q6	1	0	1	1	3/4
Q7	0	0	0	0	0/4
Q8	0	1	0	0	1/4

An analysis of Table 1 demonstrates variations in performance across both individual evaluation questions and specific rules. It is noteworthy that, in the case of Q1 (the question of whether the overall step-by-step reasoning was correct), all evaluators gave a positive response for each rule. Despite the limited sample size,

[6]While this approach is reasonable, future work may benefit from a more nuanced adjudication strategy to deal with persistent ambiguity or close splits.

the observed consistency suggests that the pipeline's structured reasoning effectively guides the LLM's output. However, output errors or hallucinations can still occur, even when the step-by-step reasoning appears sound.

Conversely, performance in error detection and correction (Q3 and Q4) exhibited greater variability. The LLM correctly identified an error in only one of the two applicable cases and was unsuccessful in correcting either of them. Following the syntax check (Q5), it was determined that rule 172 was the only one that was deemed fully correct. However, three out of four rules were judged to be faithful to the source text (Q6), indicating a generally reliable transformation from natural language to LE.

It is noteworthy that evaluators reported no instances of hallucinations (Q2) and no occurrences of extraneous information being introduced into the rule translations (Q7). However, in one instance, pertinent content was omitted (Q8), underscoring the necessity for continuous refinement.

Finally, it is imperative to emphasise that these findings are preliminary. The restricted number of rules and reviewers prevents us from making any claim of statistical significance. Nevertheless, the findings offer encouraging indications of the pipeline's prospective efficacy and establish a framework for comprehensive assessment utilising a more extensive dataset.

3.2.2 Assessment with Scenarios and Simulations. The generated rules are validated against scenarios, which are manually constructed, and executed in simulators. We briefly discuss these, though more detail is out of scope (also cf. [22]). Scenarios are modelled in Logical English by grounding the relevant predicates, then adding a query to satisfy, following the method in 2. The knowledge base can be executed against different scenarios. In this way, the function of the rules can be evaluated. Thus, scenarios can be construed as unit tests of the knowledge base. For simplicity, the scenarios are temporally punctual, as the analysis does not address temporal continuity.

For example, consider the rule in 4 - whether the vehicle "ego" moves off depends on whether or not a safe gap exists in traffic. If there is a safe gap, then the vehicle moves off; otherwise not. More complex examples can be generated, e.g., to deal with situations where the vehicle responds to an emergency vehicle.

The knowledge base can further be assessed through a simulation, by analysing the effect of the rules dynamically. Currently the NetLogo simulation shows autonomous vehicles with different setup configurations (e.g., simulated by changing the reaction time, and the desire to bend rules). This can introduce some uncertainty in the system and show how the agents would behave. This simulation can be used to evaluate the rule-base on the basis of objective metrics gathered from the running system.

We can gather metrics from the simulated system, such as time spent stopped, number of accidents, and so on. This can give important insights on the application of rules such as situations that may require additional specification in cases where the behaviour we see from AVs is not the expected one according to human judgement.

4 Related Work

As previously mentioned, recent development in autonomous mobility are making AVs a possibility in the near future. This has brought to the forefront the issue of safety, and legal compliance of these autonomous systems, especially if we consider that vulnerable users (e.g., pedestrians, cyclists, etc) will be sharing the same space.

For this reason scholars have started to consider digitising traffic laws into computable languages, albeit with different goals and implementations. There have been multiple works that deal with the formalisation of traffic rules.

We can identify two major directions in these research efforts: scenario testing and general rule representation. On one side there is research that deals mainly with specific scenarios, generally hand selected as potentially problematic, often representing the relevant rules with temporal logic. The main goal in this case is to deal with the physical aspect of driving, to ensure the correctness of the models[7]. Instances of this approach can be related to intersections [17, 21], overtaking and safe distance calculations [9], and often use languages for formal verification such as Isabelle/HOL [21]. This approach is often proposed for use in trajectory monitoring, planning, to ensure the mathematical reasoning is safe and consistent.

Another approach is to model a larger set of rules in logic form, addressing issues such as exceptions, vagueness, and ambiguity. Relevant instances of this approach use defeasible deontic logic (DDL)[5] to handle rule exceptions and the resolution of vague terms in rules. Furthermore, we could consider Prolog representations of intentions and actions[7], focusing on modelling the rules as combinations of beliefs, intentions, and context.

To model complex systems such as the road, context is necessary, which is often implicit. For this reason, the systems can be enhanced with an ontology containing additional information on driving scenarios and behaviour. The issue of integrating commonsense reasoning to fill in gaps present in the explicit traffic rules or in the surrounding world view, is also the subject of research in modelling traffic rules in ASP[14]. The converted rules can then be used in a BDI (Belief-Desire-Intention) agent, a model composed by information about the world (Beliefs), the possible states the agent wants to reach (Desires), and the selected events (Intentions)[6]. In this environment it is possible to trace the behaviour of the autonomous vehicle, and validate the rules themselves[1].

One thing that many of the above mentioned approaches have in common is the ability to be used by the autonomous agent to determine if a specific behaviour is correct (i.e., lawful). Where they can differ is if this validation happens in real time or after the fact. In the latter case the output is not used directly by the vehicle while driving, but it could be used to evaluate or further train the ML models that drive the vehicle, and evaluate the possibility of altering the rules to ensure the correct behaviour.

With regards to the issue of converting natural language rules to logic, there have been different attempts in the past (see Section 1). Recently these efforts moved towards deep learning[19] and LLMs, with recent developments also in the AV domain[18].

5 Conclusion

In this paper we presented a modular framework for translating natural language legal rules into a format that is both accessible

[7]This approach is similar to system verification in other domains, such as the railway [26]

and interpretable by humans and machines. We focus on traffic regulations in the UK Highway Code. To translate, we exploit LLMs to interpret unstructured natural language.

The goal of this work is to model traffic rules in a way that enables autonomous vehicles (AVs) to adapt their behaviour to HVs and for HVs to have normative expectations of AVs. We encode HC rules in LE, maintaining a structure as close as possible to the original legal text. The LE expressions are suitable for computation. In addition, we take a first step towards validation, developing a test framework based on a syntax checker that takes advantage of LE. Our experiments show that LLMs can be prompted to iteratively refine their translations by responding to syntax errors. This automated validation is further strengthened by running scenarios and simulations, which provide performance metrics.

Our promising results suggest that the system can effectively use step-by-step reasoning and error message feedback to generate reliable rule representations. Challenges remain, particularly in dealing with the variability in the LLM's output.

While this paper focuses on the core translation framework, several avenues for future research remain unexplored. These include: refining natural language representations of rule priorities, exceptions, and their logical formulations; more complex natural language expressions; and analysis of rule violations and unexpected behaviour in AV decision making.

Acknowledgments

The work has been partially supported by the "CompuLaw" project, funded by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant Agreement No. 833647).

References

[1] Gleifer Vaz Alves, Louise Dennis, and Michael Fisher. 2021. A Double-Level Model Checking Approach for an Agent-Based Autonomous Vehicle and Road Junction Regulations. *Journal of Sensor and Actuator Networks* 10, 3 (Sept. 2021), 41. doi:10.3390/jsan10030041

[2] Shahin Atakishiyev, Mohammad Salameh, Hengshuai Yao, and Randy Goebel. 2024. Explainable Artificial Intelligence for Autonomous Driving: A Comprehensive Overview and Field Guide for Future Research Directions. *IEEE Access* 12 (2024), 101603–101625. doi:10.1109/access.2024.3431437

[3] Avron Barr, Edward A Feigenbaum, and Paul R Cohen. 1981. *The handbook of artificial intelligence.* Vol. 3. HeurisTech Press.

[4] G. Bharathi Mohan, R. Prasanna Kumar, P. Vishal Krishh, A. Keerthinathan, G. Lavanya, Meka Kavya Uma Meghana, Sheba Sulthana, and Srinath Doss. 2024. Correction: An analysis of large language models: their impact and potential applications. *Knowledge and Information Systems* 66, 11 (July 2024), 7163–7164. doi:10.1007/s10115-024-02157-9

[5] Hanif Bhuiyan, Guido Governatori, Andry Rakotonirainy, Meng Weng Wong, and Avishkar Mahajan. 2023. Driving Decision Making of Autonomous Vehicle According to Queensland Overtaking Traffic Rules. *The Review of Socionetwork Strategies* 17, 2 (Oct. 2023), 233–254. doi:10.1007/s12626-023-00147-x

[6] Michael Bratman. 1987. *Intention, Plans, and Practical Reason.* Cambridge, MA: Harvard University Press, Cambridge.

[7] Joe Collenette, Louise A. Dennis, and Michael Fisher. 2022. Advising Autonomous Cars about the Rules of the Road. *Electronic Proceedings in Theoretical Computer Science* 371 (Sept. 2022), 62–76. doi:10.4204/EPTCS.371.5 arXiv:2209.14035 [cs]

[8] Giuseppe Contissa. [n. d.]. Rulebase Technology and Legal Knowledge Representation. In *Computable Models of the Law* (Berlin, Heidelberg, 2008), Pompeu Casanovas, Giovanni Sartor, Núria Casellas, and Rossella Rubino (Eds.). Springer, 254–262. doi:10.1007/978-3-540-85569-9_16

[9] Dan M. Costescu. 2019. Autonomous vehicles' safety in mixed traffic: Accounting for incoming vehicles when overtaking. In *2019 Electric Vehicles International Conference (EV).* 1–5. doi:10.1109/EV.2019.8893110

[10] Norbert E. Fuchs, Kaarel Kaljurand, and Tobias Kuhn. 2008. Attempto Controlled English for Knowledge Representation. In *Reasoning Web, 4th International Summer School 2008, Venice, Italy, September 7-11, 2008, Tutorial Lectures (Lecture Notes in Computer Science, Vol. 5224)*, Cristina Baroglio, Piero A. Bonatti, Jan Maluszynski, Massimo Marchiori, Axel Polleres, and Sebastian Schaffert (Eds.). Springer, 104–124. doi:10.1007/978-3-540-85658-0_3

[11] Norbert E. Fuchs, Kaarel Kaljurand, and Tobias Kuhn. 2008. Attempto Controlled English for Knowledge Representation. Springer, Berlin, Heidelberg, 104–124. doi:10.1007/978-3-540-85658-0_3

[12] Wachara Fungwacharakorn, Nguyen Ha Thanh, May Myo Zin, and Ken Satoh. 2024. Layer-of-Thoughts Prompting (LoT): Leveraging LLM-Based Retrieval with Constraint Hierarchies. arXiv:2410.12153 [cs.CL] https://arxiv.org/abs/2410.12153

[13] Richard I. Kittredge. 2022. 454Sublanguages and Controlled Languages. In *The Oxford Handbook of Computational Linguistics.* Oxford University Press. doi:10.1093/oxfordhb/9780199573691.013.015

[14] Suraj Kothawade, Vinaya Khandelwal, Kinjal Basu, Huaduo Wang, and Gopal Gupta. 2021. AUTO-DISCERN: Autonomous Driving Using Common Sense Reasoning *(CEUR Workshop Proceedings, Vol. 2970)*, Joaquin Arias, Fabio Aurelio D'Asaro, Abeer Dyoub, Gopal Gupta, Markus Hecher, Emily LeBlanc, Rafael Peñaloza, Elmer Salazar, Ari Saptawijaya, Felix Weitkämper, and Jessica Zangari (Eds.). CEUR, Porto, Portugal (virtual). https://ceur-ws.org/Vol-2970/#gdepaper7

[15] Robert Kowalski and Akber Datto. 2022. Logical English meets legal English for swaps and derivatives. *Artif Intell Law* 30 (2022), 163–197. doi:10.1007/s10506-021-09295-3

[16] Xiaohan Ma, Wenhao Yu, Chengxiang Zhao, Changjun Wang, Wenhui Zhou, Guangming Zhao, Mingyue Ma, Weida Wang, Lin Yang, Rui Mu, Hong Wang, and Jun Li. 2024. Legal Decision-Making for Highway Automated Driving. *IEEE Transactions on Intelligent Vehicles* 9, 8 (2024), 5284–5298. doi:10.1109/TIV.2023.3318214

[17] Sebastian Maierhofer, Paul Moosbrugger, and Matthias Althoff. 2022. Formalization of Intersection Traffic Rules in Temporal Logic. In *2022 IEEE Intelligent Vehicles Symposium (IV).* 1135–1144. doi:10.1109/IV51971.2022.9827153

[18] Kumar Manas, Stefan Zwicklbauer, and Adrian Paschke. 2024. TR2MTL: LLM based framework for Metric Temporal Logic Formalization of Traffic Rules. In *2024 IEEE Intelligent Vehicles Symposium (IV).* IEEE, 1206–1213. doi:10.1109/iv55156.2024.10588650

[19] Ha-Thanh Nguyen, Francesca Toni, Kostas Stathis, and Ken Satoh. 2023. Beyond Logic Programming for Legal Reasoning. arXiv:2306.16632 [cs.LO]

[20] Henry Prakken. 2017. On the problem of making autonomous vehicles conform to traffic law. *Artificial Intelligence and Law* 25, 3 (Sept. 2017), 341–363. doi:10.1007/s10506-017-9210-0

[21] Albert Rizaldi, Jonas Keinholz, Monika Huber, Jochen Feldle, Fabian Immler, Matthias Althoff, Eric Hilgendorf, and Tobias Nipkow. 2017. Formalising and Monitoring Traffic Rules for Autonomous Vehicles in Isabelle/HOL. In *Integrated Formal Methods (Lecture Notes in Computer Science)*, Nadia Polikarpova and Steve Schneider (Eds.). Springer International Publishing, Cham, 50–66. doi:10.1007/978-3-319-66845-1_4

[22] Galileo Sartor, Adam Wyner, and Giuseppe Contissa. 2024. Mind the Gaps: Logical English, Prolog, and Multi-agent Systems for Autonomous Vehicles, In Proceedings 40th International Conference on Logic Programming. *EPTCS*, 111–124. https://cgi.cse.unsw.edu.au/~eptcs/paper.cgi?ICLP2024:10

[23] Marek J. Sergot, Fariba Sadri, Robert A. Kowalski, Frank Kriwaczek, Peter Hammond, and H Terese Cory. 1986. The British Nationality Act as a logic program. *Commun. ACM* 29, 5 (1986), 370–386.

[24] Dmitry Solomakhin, Enrico Franconi, and Alessandro Mosca. 2013. Logic-based reasoning support for SBVR. *Fundamenta Informaticae* 124, 4 (2013), 543–560.

[25] LLM Stats. 2025. LLM Stats - Comparing Large Language Models. https://llm-stats.com/ Accessed: 2025-01-06.

[26] Maurice H. ter Beek. 2024. Formal Methods and Tools Applied in the Railway Domain. In *Rigorous State-Based Methods*, Silvia Bonfanti, Angelo Gargantini, Michael Leuschel, Elvinia Riccobene, and Patrizia Scandurra (Eds.). Springer Nature Switzerland, Cham, 3–21.

[27] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2024. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems* (New Orleans, LA, USA) *(NIPS '22).* Curran Associates Inc., Red Hook, NY, USA, Article 1800, 14 pages.

[28] Adam Wyner, Johan Bos, Valerio Basile, and Paulo Quaresma. 2012. An empirical approach to the semantic representation of laws. In *Legal Knowledge and Information Systems.* IOS Press, 177–180.

[29] Adam Z. Wyner. 2015. From the Language of Legislation to Executable Logic Programs. In *Logic in the Theory and Practice of Lawmaking*, Michal Araszkiewicz and Krzysztof Pleszka (Eds.). Legisprudence Library, Vol. 2. Springer, 409–434. doi:10.1007/978-3-319-19575-9_15

[30] Adam Z. Wyner and Wim Peters. 2011. On Rule Extraction from Regulations. In *Legal Knowledge and Information Systems - JURIX 2011: The Twenty-Fourth Annual Conference, University of Vienna, Austria, 14th-16th December 2011 (Frontiers in*

Artificial Intelligence and Applications, Vol. 235), Katie Atkinson (Ed.). IOS Press, 113–122. doi:10.3233/978-1-60750-981-3-113

[31] Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, Yang Wang, and Enhong Chen. 2024. Large language models for generative information extraction: a survey. *Frontiers of Computer Science* 18, 6 (Nov. 2024). doi:10.1007/s11704-024-40555-y

[32] Daniela Zambrini and Orlando Chiarello. 2023. Subject Fields in a Controlled Natural Language: How the Evolution of the ASD-STE100 Specification Led to a Proposal for a Global Structured Review of Term Categories (Short Paper). *Proceedings http://ceur-ws. org ISSN* 1613 (2023), 0073.