SEMANTIC ROLE-ASSISTED NATURAL LANGUAGE RULE FORMALIZATION FOR INTELLIGENT VEHICLE

Kumar Manas[1,2] & Adrian Paschke[1,3]

1. Freie University of Berlin, Germany

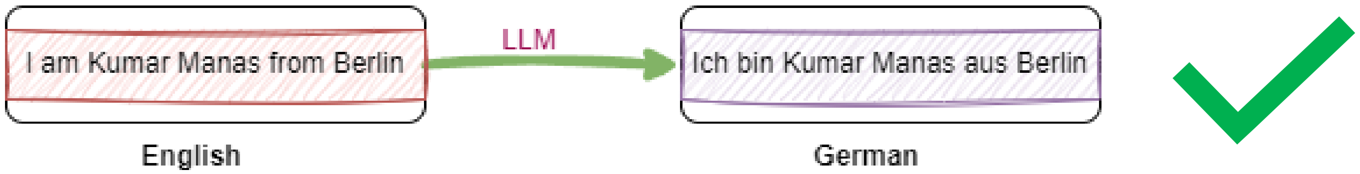
2. Continental AG

3. Fraunhofer Institute for Open Communication Systems, Berlin

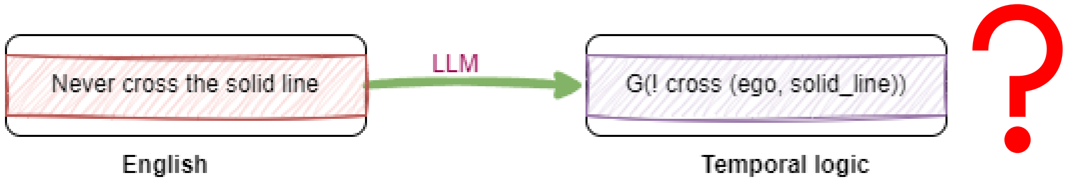
THE 7TH INTERNATIONAL JOINT CONFERENCE ON RULES AND REASONING
Oslo, Norway 18 - 20 September 2023

September 19, 2023

MOTIVATION



I am Kumar Manas from Berlin
LLM
Ich bin Kumar Manas aus Berlin
English
German

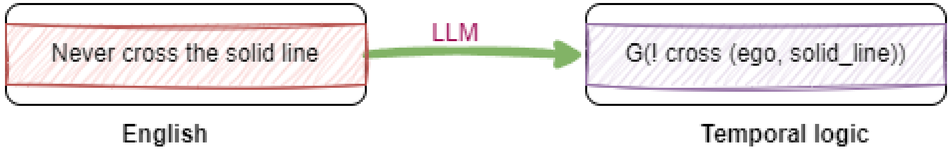


Never cross the solid line
LLM
G(! cross (ego, solid_line))
English
Temporal logic

SEMANTIC PARSING

- Task of converting a natural language utterance to a logical form
  - machine-understandable representation of text.

- Natural language to Metric Temporal Logic
  - A semantic Parsing Task



LINEAR TEMPORAL LOGIC (LTL)

- LTL grammar:

$$\varphi ::= p \mid \neg p \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2$$

Where, $p \in P$, set of possible atomic proposition

φ is task specification

$\varphi_1 and\ \varphi_2\ are\ LTL\ formula$

LINEAR TEMPORAL LOGIC (LTL)

- LTL and Temporal operators:

$$\varphi ::= G(\varphi) \mid \varphi_1 U \varphi_2 \mid X(\varphi) \mid F(\varphi) \mid P(\varphi)$$

G, U, X, F, P is temporal operator.

METRIC TEMPORAL LOGIC (MTL)

- MTL extends linear temporal logic by adding time constraints to the modal operators, making it more expressive.
- MTL supports past and future with precise timing constraints, effectively resolving traffic situations.

- MTL grammar:

$$\varphi ::= p \mid \neg p \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2$$

Where, $p \in P$, set of possible atomic proposition

φ is task specification

$\varphi_1 and\ \varphi_2\ are\ MTL\ formula$

METRIC TEMPORAL LOGIC (MTL)

- MTL and Temporal operators:

$$\varphi ::= G_t(\varphi) \mid \varphi_1 U_t \varphi_2 \mid X_t(\varphi) \mid F_t(\varphi) \mid P_t(\varphi)$$

G, U, X, F, P is temporal operator.

t represents time interval $[t_1, t_2]$

TRAFFIC RULE: DATASET SNAPSHOT

No public available dataset for MTL.

We created traffic rules and MTL pairs.

35 such pairs created with very complex rules.

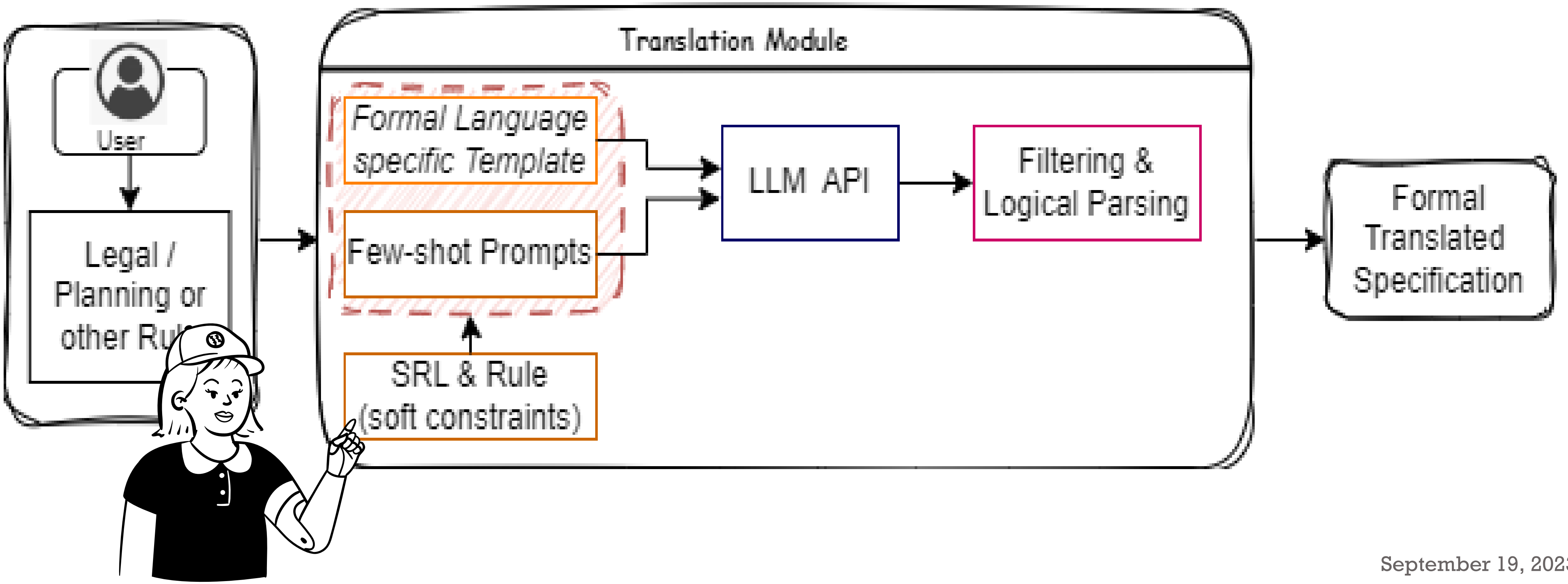
Source: German and UN traffic rules

TRAFFIC RULE: DATASET SNAPSHOT

If ego vehicle wants to change lanes, turn, or overtake, they should use their turn signals beforehand for t time units.

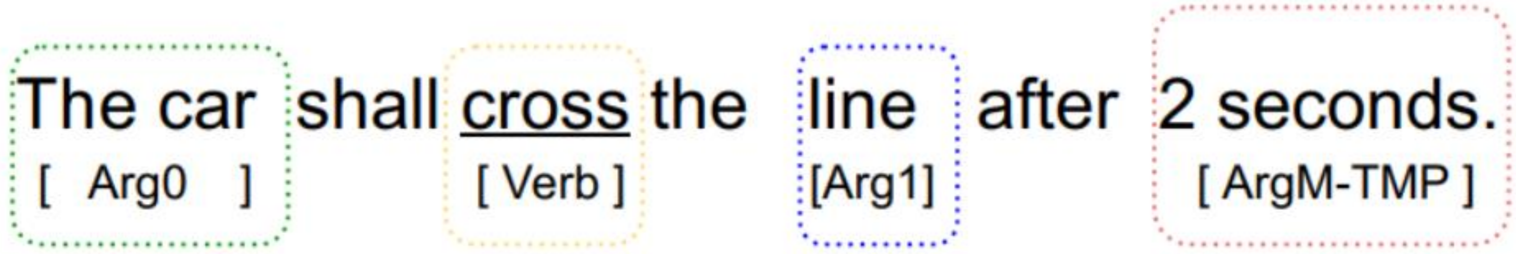
$$G\Big(\mathrm{P}_{[0,t]}\mathrm{turn\_signal(ego)} \Rightarrow \big(\mathrm{change\_lane(ego)} \vee \mathrm{turn(ego)} \vee \mathrm{overtake(ego)}\big)\Big)$$

MODEL - ARCHITECTURE



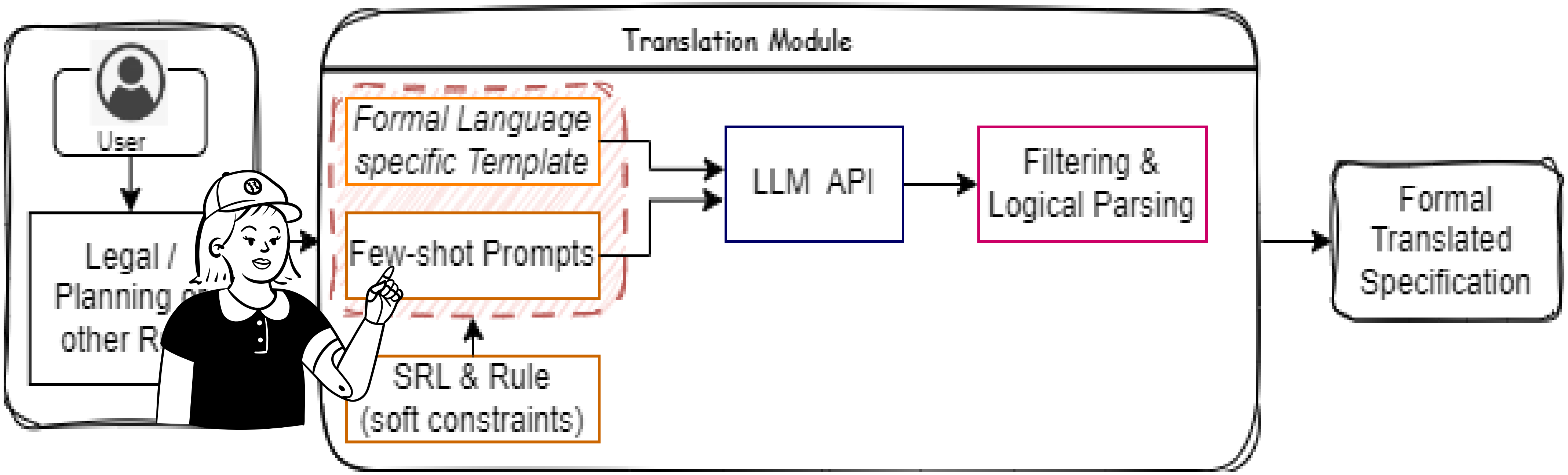
SEMANTIC ROLE LABEL (SRL)

- *„who did what to whom, and how, when and where“ [1]*



- *VerbNet[1] & PropBank[2]* are two paradigm for SRL.

MODEL - ARCHITECTURE



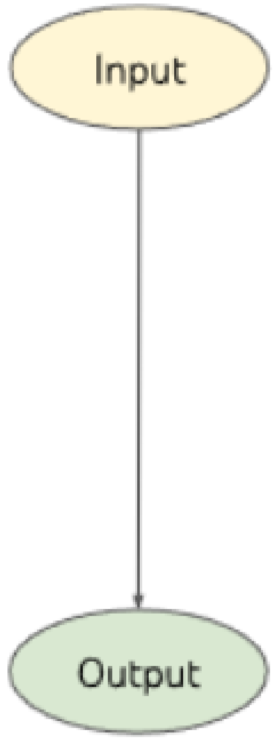
FEW-SHOT PROMPTING

Two ways to use the language model (chatGPT, Bard…..)

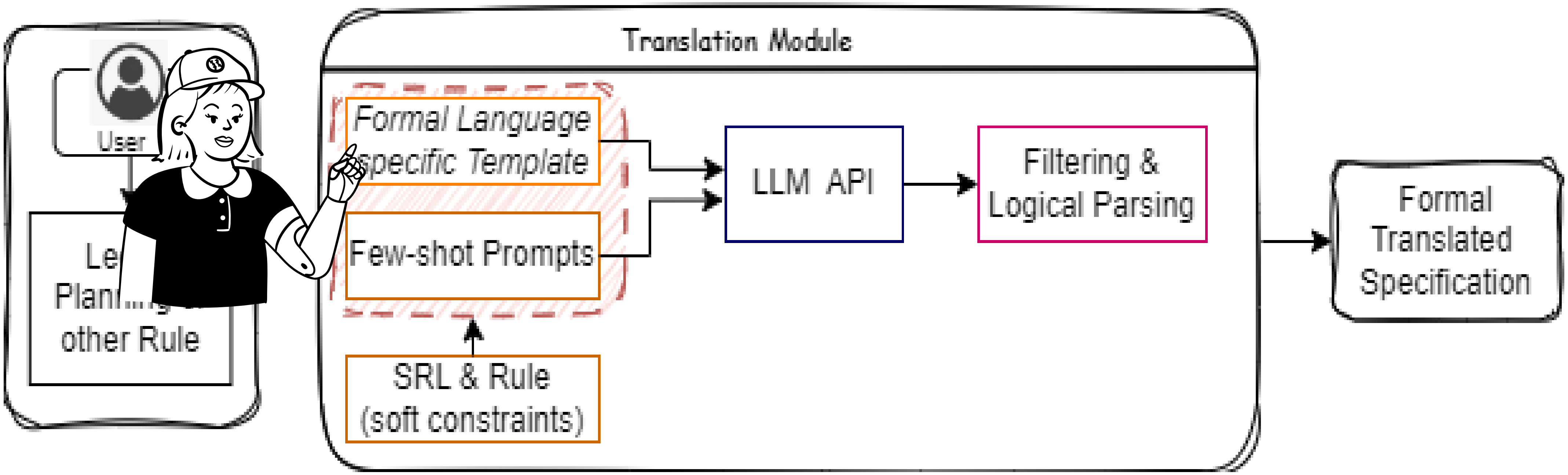
- Fine-tuning
- Prompting

Prompting: Provide input to model to generate output.

Few-shot Prompting: Model provided with „N“ example pairs to replicate similar action during test time translation.



MODEL - ARCHITECTURE



COMPLETE LLM INPUT

Translate the following natural language traffic rule into MTL formula based on the semantic role label (SRL) information and rule as soft constraints, and think step-by-step using chain-of-thought.
The MTL formula should contain temporal and logical operators |, &, ~, ->, <->, X, I, G, F, O, P, U, and if the interval is not specified for a temporal operator then its till the end.

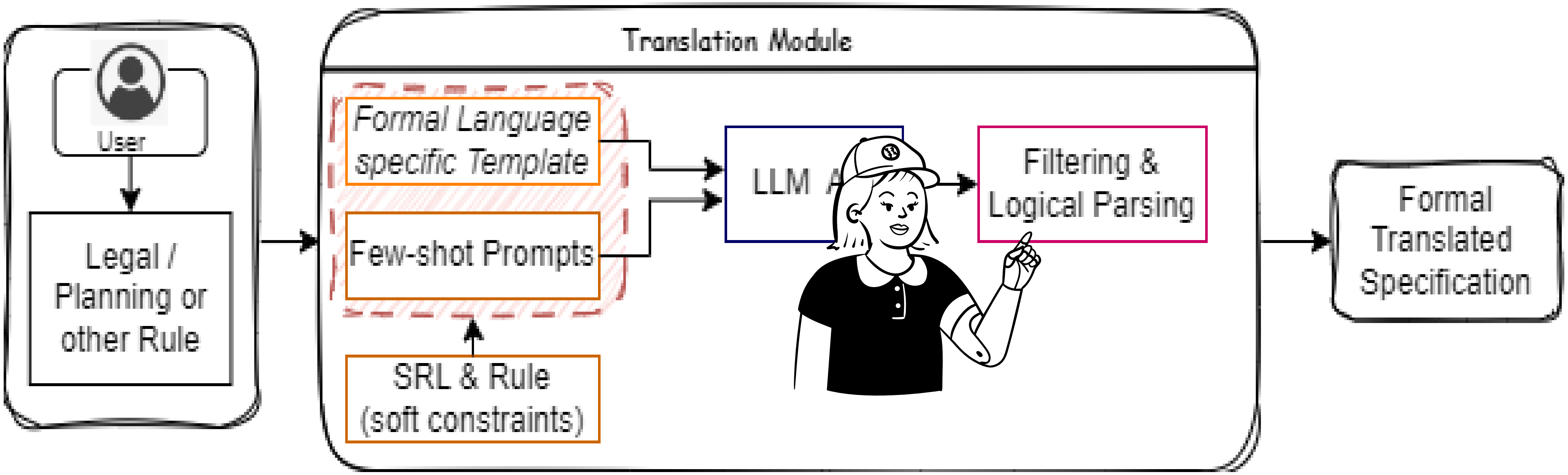
1

Natural language traffic rule: Ego vehicle will not exceed the speed limit of the lane it is driving on, and it will not exceed the maximum velocity allowed for its vehicle type, and ego will not exceed the speed limit such that it can no longer react to traffic regulations and restrictions.
Additional SRL and Rule Knowledge: ''Obtained from external pre-trained SRL module and soft rule constraints.''
MTL translation: G(keep_lane_speed_limit(ego) ∧ keep_vehicle_type_limit(ego) ∧ keep_braking_speed_limit(ego)).

2

Natural language traffic rule: If an ego vehicle wants to overtake the other vehicle, then it should use turn signals beforehand for t seconds.
Additional SRL and Rule Knowledge: ''Obtained from external pre-trained SRL module and soft rule constraints.''
MTL translation: G(overtake(ego,other) -> P[0,t] turn_signal(ego)).

MODEL - ARCHITECTURE



FILTERING & LOGICAL PARSING

Language models hallucinate.

Little idea about formal logic, unlike mathematical or coding understanding.

LTLf2DFA & PyMTL: Temporal logic parser to check the model output.

Is output grammatically correct?

More trust for end users ☺

ALGORITHM: RULES TO FORMAL LOGIC

procedure CONVERTTEXTUALRULESTOTEMPORALLOGIC(*TextualRuleDataset*)
    results ← {}
    for all *rule* ∈ *TextualRuleDataset* **do**
        semanticRoleofWords ← *semanticRoleLabelingModule*(*rule*)
        additionalKnowledge ← *userDefinedRule*(*semanticRoleofWords*)
        intermediateTranslation ← *LLM*(*rule* + *additionalKnowkedge*)
        temporalLogicTranslation ← *LLM*(*intermediateTranslation*)
        LogicallyParsedTranslation ← *parser*(*temporalLogicTranslation*)
        if *LogicallyParsedTranslation* ≠ *error* **then**
            results.append(*LogicallyParsedTranslation*)
        else
            results.append(*temporalLogicTranslation*)
        end if
    end for
    finalTemporalLogic ← *majorityVote*(*results*)
    return *finalTemporalLogic*
end procedure

ALGORITHM: RULES TO FORMAL LOGIC

```
procedure ConvertTextualRulesToTemporalLogic(TextualRuleDataset)
    results ← {}
    for all rule ∈ TextualRuleDataset do
        semanticRoleofWords ← semanticRoleLabelingModule(rule)
        additionalKnowledge ← userDefinedRule(semanticRoleofWords)
        intermediateTranslation ← LLM(rule + additionalKnowkedge)
        temporalLogicTranslation ← LLM(intermediateTranslation)
        LogicallyParsedTranslation ← parser(temporalLogicTranslation)
        if LogicallyParsedTranslation ≠ error then
            results.append(LogicallyParsedTranslation)
        else
            results.append(temporalLogicTranslation)
        end if
    end for
    finalTemporalLogic ← majorityVote(results)
    return finalTemporalLogic
end procedure
```

ALGORITHM: RULES TO FORMAL LOGIC

```
procedure ConvertTextualRulesToTemporalLogic(TextualRuleDataset)
    results ← {}
    for all rule ∈ TextualRuleDataset do
        semanticRoleofWords ← semanticRoleLabelingModule(rule)
        additionalKnowledge ← userDefinedRule(semanticRoleofWords)
        intermediateTranslation ← LLM(rule + additionalKnowkedge)
        temporalLogicTranslation ← LLM(intermediateTranslation)
        LogicallyParsedTranslation ← parser(temporalLogicTranslation)
        if LogicallyParsedTranslation ≠ error then
            results.append(LogicallyParsedTranslation)
        else
            results.append(temporalLogicTranslation)
        end if
    end for
    finalTemporalLogic ← majorityVote(results)
    return finalTemporalLogic
end procedure
```

ALGORITHM: RULES TO FORMAL LOGIC

```
procedure ConvertTextualRulesToTemporalLogic(TextualRuleDataset)
    results ← {}
    for all rule ∈ TextualRuleDataset do
        semanticRoleofWords ← semanticRoleLabelingModule(rule)
        additionalKnowledge ← userDefinedRule(semanticRoleofWords)
        intermediateTranslation ← LLM(rule + additionalKnowkedge)
        temporalLogicTranslation ← LLM(intermediateTranslation)
        LogicallyParsedTranslation ← parser(temporalLogicTranslation)
        if LogicallyParsedTranslation ≠ error then
            results.append(LogicallyParsedTranslation)
        else
            results.append(temporalLogicTranslation)
        end if
    end for
    finalTemporalLogic ← majorityVote(results)
    return finalTemporalLogic
end procedure
```

EVALUATION: TRAFFIC RULE-MTL

Model Architecture with LLM Backbone	Without SRL	With SRL (PropBank)	With SRL (VerbNet)
Ours (GPT-3.5-turbo [3])	37.50%	**56.25%**	**53.13%**
Ours (Text-davinci-003[3])	**40.60%**	53.13%	**53.13%**
Ours (StarCoder [4])	12.50%	18.75%	18.75%
Ours (Bloomz [5])	9.38%	12.50%	15.62%
nl2ltl [6]	28.15%	-	-

EVALUATION: TRAFFIC RULE-MTL

Model Architecture with LLM Backbone	Without SRL	With SRL (PropBank)	With SRL (VerbNet)
Ours (GPT-3.5-turbo [3])	37.50%	**56.25%**	**53.13%**
Ours (Text-davinci-003[3])	**40.60%**	53.13%	**53.13%**
Ours (StarCoder [4])	12.50%	18.75%	18.75%
Ours (Bloomz [5])	9.38%	12.50%	15.62%
nl2ltl [6]	28.15%	-	-

EVALUATION: DRONE SPECS-LTL

- Drone specification dataset [7] for planning of drone through LTL.
- 5900 drone instruction-LTL pairs with 5 unique logical structure

Model backbone	Fine-tuning	Prompting	Acccuracy
Ours (SRL-PropBank with GPT-3.5-turbo [3])	✗	✓	44.06%
Ours (SRL-VerbNet with GPT-3.5-turbo [3])	✗	✓	46.00%
Ours (without SRL GPT-3.5-turbo [3])	✗	✓	40.01%
BART-FT-RAW [7]	✓	✗	**69.39%**

CONCLUSION

- Language models can perform semantic parsing or translation of text into formal logic.
- Accuracy still not great for safety-critical systems.
- Good News ☺ SRL alongside rule as soft constraints can reduce the data need.
  - Data collection costs money and reduces user reachability.
- SRL is fading, but they showed that we should hold them with a subsymbolic language models.

REFERENCES

[1] Palmer, M., Gildea, D., Xue, N.: Semantic Role Labeling. Synthesis Lectures on Human Language Technologies, Springer International Publishing, Cham (2010)

[2] Palmer, M., Gildea, D., Kingsbury, P.: The Proposition Bank: An annotated corpus of semantic roles. Computational Linguistics 31(1), 71–106 (2005).

[3] https://platform.openai.com/docs/models/gpt-3-5

[4] Li, R., Allal, L.B., Zi, Y., Muennighoff, N., Kocetkov, D., Mou, C., Marone, M., Akiki, C., Li, J., Chim, J., et al.: Starcoder: may the source be with you! ArXiv abs/2305.06161 (2023)

[5] Muennighoff, N., Wang, T., Sutawika, L., Roberts, A., Biderman, S.R., Scao, T.L., Bari, M.S., et al.: Crosslingual generalization through multitask finetuning. ArXiv abs/2211.01786 (2022)

[6] Fuggitti, F., Chakraborti, T.: NL2LTL – a python package for converting natural language (NL) instructions to linear temporal logic (LTL) formulas. In: AAAI (2023), system Demonstration.

[7] Pan, J., Chou, G., Berenson, D.: Data-efficient learning of natural language to linear temporal logic translators for robot task specification. arXiv e-prints pp. arXiv–2303 (2023)