BUILDING FULLY INTERPRETABLE NLG SYSTEM WITH LLMS

Jędrzej Warczyński
Poznan University of Technology, Poland

Bridging Interpretability and Performance in Data-to-Text Systems

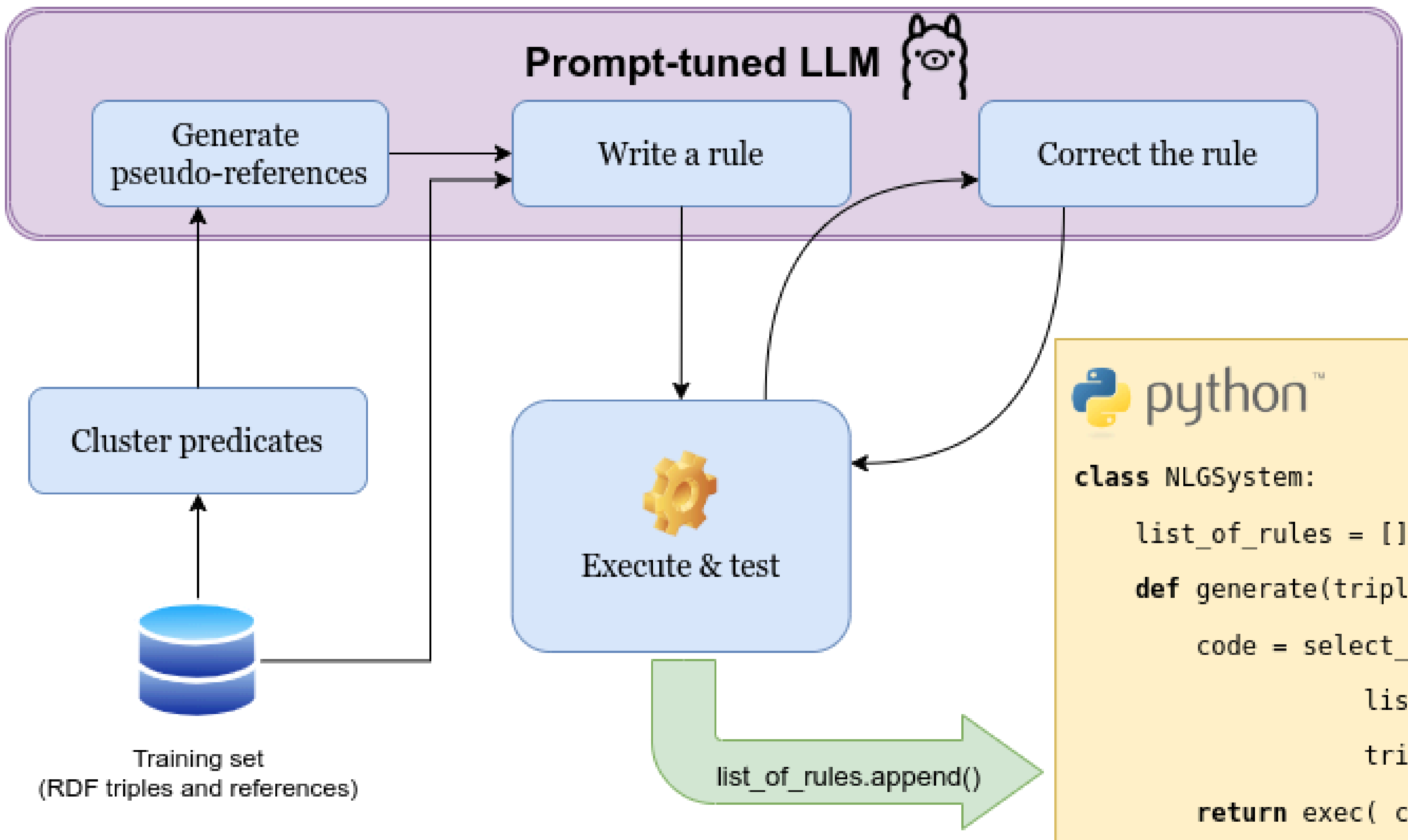
INTRODUCTION

Data-to-text is a field of natural language generation (NLG) that focuses on converting structured, non-linguistic data into coherent text

```
RDF triples:
(Mozart, birthplace, Vienna),
(Mozart, birth year, 1756)

"Mozart was born in 1756 in Vienna."
```

There are two main approaches to the construction of data-to-text systems: rule-based and neural methods



python

```python
class NLGSystem:
    list_of_rules = []
    def generate(triples):
        code = select_rule(
                    list_of_rules,
                    triples )
        return exec( code,
                     triples )
```

MOTIVATION

- Interpretability
- Computational performance

IDEA

- Combining deep neural network & rule-based perspectives on building NLG systems
- Using a large language model to **write** a rule-based system in *pure Python*

TRAINING PROCEDURE

- Processes the training set by asking a large language model to write simple Python code that would generate the reference text based on the input data.
- The generated code is executed to check for syntax errors and whether it produces the correct output.
- The final result of the training of the system is a single file of Python code that is able to generate the textualisation for the input data.

RESULTS

- Rule-based approach ranked second in both the BLEU and BLEURT metrics
- Rule-based outperforms the prompt-tuned Llama 3 70B model
- Our approach generates texts on a single CPU 83 times faster than the fastest neural approach (BART) running on a GPU

	BLEU	METEOR	BLEURT	TIME GPU	TIME CPU	Interpretability
promp-tuned LLM	38.26	0.680	0.113	1h 46min	n/a	✗
rule-based (ours)	42.51	0.671	0.157	-	**3s**	✓
fine-tuned BART	**53.28**	**0.716**	**0.257**	249s	1910s	✗

SUMMARY

- Extremely **fast**
- Fully **interpretable**
- **Similar quality** to another approaches
- Does not allow the generation of rules for the unknown, i.e. out-of-domain predicates ?

HUMAN EVALUATION

- 5 annotators, 75 test instances, 225 systems outputs
- lowest number of minor hallucinations (typos in named entity names)
- lowest number of disfluencies
- lowest number of repetitions
- significantly fewer major hallucinations (output containing facts not supported by the data) than fine-tuned BART

	min. hal.	maj. hal.	omissions	disfluencies	repetitions
promp-tuned LLM	0.08	**0.07**	**0.07**	0.19	**0.03**
rule-based (ours)	**0.04**	0.013	0.08	**0.13**	**0.03**
fine-tuned BART	0.20	0.33	**0.19**	0.16	0.07