Data ScienceTech Institute

Investigating Language Model Capabilities to Represent and Process Formal Knowledge: A Preliminary Study to Assist Ontology Engineering

Hanna Abi Akl

Supervised by: Fabien Gandon, Catherine Faron and Pierre Monnin

Doctoral Consortium

Content

About My Work

- Joint PhD student between Data ScienceTech Institute (DSTI) and INRIA France
- PhD thesis: *Neuro-symbolic reasoning in Language Models to bootstrap Ontology Engineering*
- Goal: Establish Language Models (LMs) as effective assistants in ontology extension
- Current limitations:
  - Ontology extension is manual and time-consuming
  - Natural language is ambiguous for representing information
  - Language models hallucinate
  - Language models are black-boxes

Research Objectives

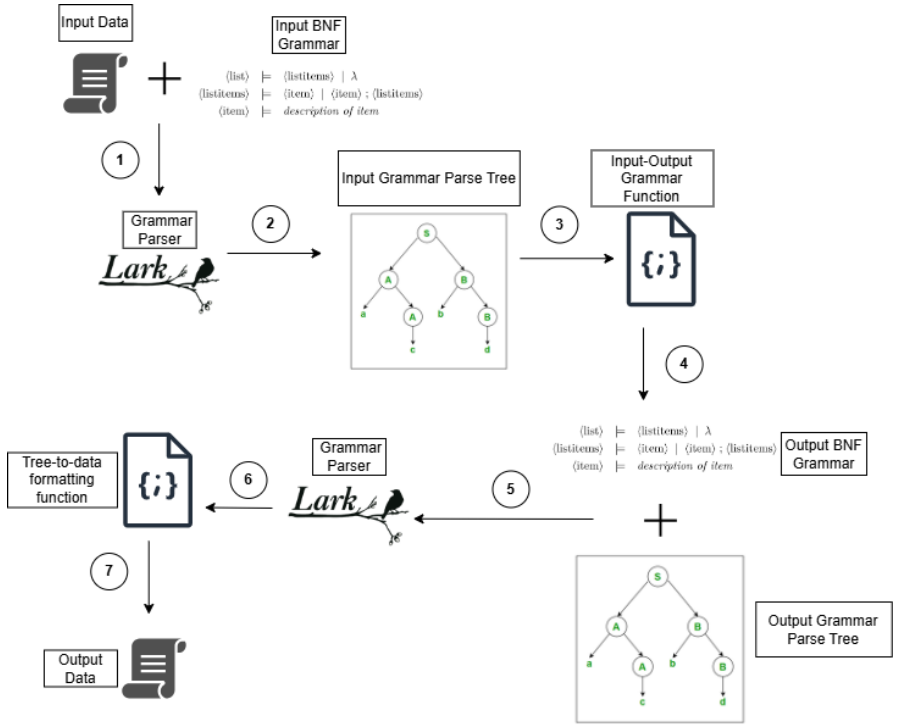
- Combine LM knowledge with symbolic methods for ontological tasks
- Refine LMs as a reliable interface to assist ontologists
- Use LMs to explain ontological choices like class extensions
- Research Questions (RQs):
  - *RQ1: How do different formal representations affect LM reasoning?*
  - *RQ2: How can LMs use formal representations to extend an ontology?*
  - *RQ3: How can LMs explain generated ontological choices with logical tools like syllogisms?*

Proposed Approach

- Identification of viable formal data language as NL alternative **(current work)**
- Efficient encoding of external knowledge in LMs for ontology extension
- Evaluation methods for LM-generated ontological classes and properties
- Injection of logical tools like syllogistic reasoning to guide and explain LM generation

Proposed Approach

- SEF-CLGC pipeline: automated generation and evaluation of different formal languages for LMs on FOL reasoning task
- Language selection criteria:
  - Verbosity → compact vs verbose
  - Frequency → seen vs unseen by LM
  - Abstractness → natural language vs mathematical
  - Representation → finiteness



RULEML+RR 2025

wimmics Inria INVENTEURS DU MONDE NUMÉRIQUE

UNIVERSITÉ CÔTE D'AZUR

DSTI

Proposed Approach

Table 1
SEF classification examples on FOLIO.

NL Premise	NL Conclusion	FOL Premise	FOL Conclusion	SEF Class
All squares are four-sided. All four-sided things are shapes.	All squares are shapes.	∀x (Square(x) ⟶ FourSided(x)) ∀x (FourSided(x) ⟶ Shape(x))	∀x (Square(x) ⟶ Shape(x))	Hypothetical
Some affection is love. Some love is positive.	Some affection is positive.	∃x (Affection(x) ∧ Love(x)) ∃x (Love(x) ∧ Positive(x))	∃x (Affection(x) ∧ Positive(x))	Categorical
Diamond Mine is a professional wrestling stable formed in WWE. Roderick Strong leads Diamond Mine. Diamond Mine includes the Creed Brothers and Ivy Nile. Imperium has a feud with Diamond Mine.	Roderick Strong leads the Creed Brothers.	ProfessionalWrestlingStable(diamondMine) ∧ In(diamondMine, WWE) Leads(roderickStrong, diamondMine) Includes(diamondMine, creedBrothers) ∧ Includes(diamondMine, ivyNile) Feuds(imperium, diamondMine)	Leads(roderickstrong, creedbrothers)	Complex
Susan flies to LGA airport. The departure and arrival can not be at the same airport. John flies from LGA airport.	Susan flies from LGA airport.	FlyTo(susan, lgaAirport) ∀x ∀y (FlyFrom(x, y) ⊕ FlyTo(x, y)) FlyFrom(john, lgaAirport)	FlyFrom(susan, lgaAirport)	Disjunctive

Table 3
Example transformations from FOL to CLGC languages.

FOL	MINIFOL	CLIF	CGIF	TFL	TFL+
∀x ((Employee(x) ∧ Schedule(x, meeting, customers)) ⟶ AppearIn(x, company))	all:x ((employee(x) & schedule(x, meeting, customers)) :- appearin(x, company))	forall x ((employee(x) and schedule(x, meeting, customers)) implies appearin(x, company))	[@every *x [([[employee[(?x)] schedule[(?x meeting customers)]]] appearin[(?x company)]]]	-+E1++S1-+A1	-((+E0++S0)-+A0)
∀x ((Employee(x) ∧ HasLunch(x, company)) ⟶ Schedule(x, meeting, customers))	all:x ((employee(x) & haslunch(x, company)) :- schedule(x, meeting, customers))	forall x ((employee(x) and haslunch(x, company)) implies schedule(x, meeting, customers))	@every *x [([[employee[(?x)] haslunch[(?x company)]] schedule[(?x meeting customers)]]]	-+E1++H1-+S1	-((+E0++H0)-+S0)
∀x (Employee(x) ⟶ (HasLunch(x, company) ⊕ HasLunch(x, home)))	all:x ((employee(x) :- (haslunch(x, company) haslunch(x, home)))	forall x (employee(x) implies (haslunch(x, company) xor haslunch(x, home)))	@every *x [(employee[(?x)] [(haslunch[(?x company)] haslunch[(?x home)])])]	-+E1-+H1-+H1	-(+E0-(+H0-+H0))
∀x ((Employee(x) ∧ HasLunch(x, home)) ⟶ Work(x, home))	all:x ((employee(x) & haslunch(x, home)) :- work(x, home))	forall x ((employee(x) and haslunch(x, home)) implies work(x, home))	@every *x [([[employee[(?x)] haslunch[(?x home)]]] work[(?x home)]]]	-+E1++H1-+W1	-((+E0++H0)-+W0)
∀x ((Employee(x) ∧ (¬In(x, homecountry))) ⟶ Work(x, home))	all:x ((employee(x) & (in(x, homecountry))) :- work(x, home))	forall x ((employee(x) and (not in(x, homecountry))) implies work(x, home))	@every *x [([[employee[(?x)] [(in[(?x homecountry)])]]] work[(?x home)]]]	-+E1+-+I1-+W1	-((+E0+(-+I0))-+W0)
∀x (Manager(x) ⟶ ¬Work(x, home))	all:x (manager(x) :- work(x, home))	forall x (manager(x) implies not work(x, home))	@every *x [(manager[(?x)] work[(?x home)]]]	-+M1-+W1	-(+M0-+W0)
¬(Manager(james) ⊕ AppearIn(james, company))	(manager(james) ∧ appearin(james, company))	not (manager(james) xor appearin(james, company))	[(manager[(james)] appearin[(james company)])]]	-+M1-+A1	-(+M2(+j2)-+A2)
HasLunch(james, company)	haslunch(james, company)	haslunch(james, company)	[haslunch[(james company)]]	+H1	+H2

Proposed Approach

- Dataset: FOLIO → set of n premises and m conclusions in NL and their corresponding FOL annotations
- Goal: Predict if conclusions are True, False or Uncertain
- Runs consist of variations of (Model, Grammar, Learning Method)
- Sub-Research Questions (SRQs):
  - *SRQ1: Which training method yields the best results for solving FOL problems with LMs?*
  - *SRQ2: How do formal representations scale with models?*
  - *SRQ3: Does a more compact vocabulary boost model performance for formal languages?*

Preliminary Results

- SFT results show that NL is still the best representation
- CLIF representation offers a competitive formal alternative
- Formal languages scale consistently with model size

Table 6
Supervised Fine-Tuning Results on A100 GPU: best results in bold and second best underlined.

Model	Grammar	Accuracy	Precision	Recall	F1
Flan-T5-base	CLIF	0.5073	0.5001	0.5015	0.4986
Flan-T5-base	NL	0.5418	0.5425	0.5397	0.5387
Flan-T5-base	TFL	0.4827	0.6240	0.4646	0.4187
Flan-T5-large	**NL**	**0.6600**	**0.6622**	**0.6572**	**0.6585**
Flan-T5-large	CLIF	0.6157	0.6148	0.6156	0.6149
Flan-T5-base	FOL	0.4876	0.4716	0.4727	0.4444
Flan-T5-base	TFL+	0.4926	0.5690	0.4775	0.4504
Flan-T5-large	TFL+	0.5418	0.5594	0.5350	0.5347

Preliminary Results

- ZS and FS training methods perform less well than SFT
- Grammar passing boosts ZS and has no effect on FS prompting
- Formal languages keep same performance order in ZS as in SFT

Table 8
Zero-Shot with and without BNF Grammar Prompting Results on L4 GPU. The best results are in bold and the second best are underlined.

Model	Grammar	Accuracy	Precision	Recall	F1	Grammar Prompting
Gemma-2-2b-it	**CLIF**	**0.4532**	0.4633	**0.4368**	**0.3924**	Yes
Gemma-2-2b-it	CLIF	0.4433	0.5028	0.4346	0.3672	No
Gemma-2-2b-it	FOL	0.4187	0.4021	0.4018	0.3514	Yes
Gemma-2-2b-it	FOL	0.4334	0.4989	0.4233	0.3574	No
Gemma-2-2b-it	TFL+	0.3596	0.3618	0.3445	0.2782	Yes
Gemma-2-2b-it	**TFL+**	0.3399	**0.5634**	0.3310	0.2498	No
Gemma-2-2b-it	TFL	0.3645	0.4689	0.3458	0.2422	Yes
Gemma-2-2b-it	TFL	0.3399	0.2225	0.2419	0.1863	No

Table 10
8-Shot with and without BNF Grammar Prompting Results on L4 GPU.

Model	Grammar	Accuracy	Precision	Recall	F1	Grammar Prompting
Gemma-2-2b-it	TFL	0.3546	0.1182	0.3333	0.1745	Yes
Gemma-2-2b-it	TFL	0.3546	0.1182	0.3333	0.1745	No
Gemma-2-2b-it	TFL+	0.3546	0.1182	0.3333	0.1745	Yes
Gemma-2-2b-it	TFL+	0.3546	0.1182	0.3333	0.1745	No
Gemma-2-2b-it	FOL	0.3546	0.1182	0.3333	0.1745	Yes
Gemma-2-2b-it	FOL	0.3546	0.1182	0.3333	0.1745	No
Gemma-2-2b-it	CLIF	0.3546	0.1182	0.3333	0.1745	Yes
Gemma-2-2b-it	CLIF	0.3546	0.1182	0.3333	0.1745	No

RULEML+RR 2025

wimmics Inria INVENTEURS DU MONDE NUMÉRIQUE

UNIVERSITÉ CÔTE D'AZUR

DSTI

Preliminary Results

- Tokenizer re-training on language grammar yields impressive performances at small scale (e.g. TFL+) but collapses with scaling
- Does not outperform standard SFT

Table 12
Supervised Fine-Tuning with Tokenizer Re-Training Results. The best results are in bold and the second best are underlined.

Model	Grammar	Accuracy	Precision	Recall	F1	GPU	Re-Train Tokenizer	Vocabulary Size
Flan-T5-small	TFL	0.3201	0.2983	0.3109	0.2790	T4/L4	Yes	191
Flan-T5-small	TFL	0.3596	0.3424	0.3476	0.3070	T4/L4	No	32128
Flan-T5-small	CLIF	0.3497	0.3600	0.3466	0.3382	T4/L4	Yes	32128
Flan-T5-small	CLIF	0.4384	0.4425	0.4274	0.4109	T4/L4	No	32128
Flan-T5-small	TFL+	0.4532	0.4286	0.4395	0.4113	T4/L4	Yes	180
Flan-T5-small	TFL+	0.3596	0.3243	0.3440	0.2918	T4/L4	No	32128
Flan-T5-base	TFL+	0.4334	0.4083	0.4168	0.3713	A100	Yes	180
Flan-T5-base	**TFL+**	0.4926	**0.5690**	0.4775	0.4504	A100	No	32128
Flan-T5-large	TFL+	0.4039	0.4786	0.3854	0.3167	A100	Yes	180
Flan-T5-large	**TFL+**	**0.5418**	0.5594	**0.5350**	**0.5347**	A100	No	32128

Preliminary Results

- SEF comparison shows that formal languages have similar performance on different syllogism categories as NL
- Formal languages like CLIF can replace NL as less verbose formalization without sacrificing too much performance

Table 13
Syllogism Evaluation Framework for Supervised Fine-Tuning Flan-T5-large on NL, CLIF and TFL+.

Grammar	Hypothetical		Disjunctive		Complex		Categorical	
	Hit	Miss	Hit	Miss	Hit	Miss	Hit	Miss
NL	59	25	60	39	14	2	2	2
CLIF	48	36	57	42	14	2	2	2
TFL+	44	40	53	46	15	1	2	2

Preliminary Results

- *SRQ1: Which training method yields the best results for solving FOL problems with LMs?* **SFT remains the best training method**
- *SRQ2: How do formal representations scale with models?* **Controlled formal languages scale well while keeping consistent performances**
- *SRQ3: Does a more compact vocabulary boost model performance for formal languages?* **Tailoring the tokenizer vocabulary to that of the formal language yields erratic performances**

Conclusion

- Summary
  - CLIF is a strong alternative for NL data representation
- Contributions
    - SEF-CLGC pipeline
    - In-context grammar passing for prompting strategy
    - Tokenizer re-training on formal language vocabulary
- Future work
  - Encode knowledge in formal language for ontology tasks
  - Generate and use syllogistic reasoning to explain LM ontology engineering choices
  - Evaluate on HMAS domain data

Research Plan

Year 1

- Literature review
- Framework
- Metrics

Year 2

- Framework
- Metrics
- Evaluation
- Results Analysis

Year 3

- Evaluation
- Results Analysis
- Domain Extension

RULEML+RR 2025

wimmics Inria INVENTEURS DU MONDE NUMÉRIQUE

UNIVERSITÉ CÔTE D'AZUR

DSTI

References

- H. Liu, Z. Fu, M. Ding, R. Ning, C. Zhang, X. Liu, Y. Zhang, Logical reasoning in large language models: A survey, Preprint arXiv:2502.09100 (2025).
- W. Wang, Y. Yang, F. Wu, Towards data-and knowledge-driven ai: a survey on neuro-symbolic computing, IEEE Transactions on Pattern Analysis and Machine Intelligence (2024).
- A. S. Lippolis, M. J. Saeedizade, R. Keskisärkkä, S. Zuppiroli, M. Ceriani, A. Gangemi, E. Blomqvist, A. G. Nuzzolese, Ontology generation using large language models, in: European Semantic Web Conference, Springer, 2025, pp. 321–341.
- Y. Zhao, Leveraging large language models for ontology requirements engineering, in: Extended Semantic Web Conference ESWC, 2025, pp. –.
- Z. Hou, Neural-symbolic reasoning: Towards the integration of logical reasoning with large language models, Authorea Preprints (2025).
- J. Huang, K. C.-C. Chang, Towards reasoning in large language models: A survey, Preprint arXiv:2212.10403 (2022). [14] G. Srivastava, S. Cao, X. Wang, Towards reasoning ability of small language models, Preprint arXiv:2502.11569 (2025).
- M. Besta, J. Barth, E. Schreiber, A. Kubicek, A. Catarino, R. Gerstenberger, P. Nyczyk, P. Iff, Y. Li, S. Houliston, et al., Reasoning language models: A blueprint, Preprint arXiv:2501.11223 (2025).
- K. Kumar, T. Ashraf, O. Thawakar, R. M. Anwer, H. Cholakkal, M. Shah, M.-H. Yang, P. H. Torr, F. S. Khan, S. Khan, Llm post-training: A deep dive into reasoning large language models, Preprint arXiv:2502.21321 (2025).

Contact

Thank you

Contact: hanna.abi-akl@inria.fr
Website: https://hannaabiakl.github.io/
GitHub: https://github.com/HannaAbiAkl

RULEML+RR 2025

wimmics Inria INVENTEURS DU MONDE NUMÉRIQUE

UNIVERSITÉ CÔTE D'AZUR

DSTI