VERIFY: A Novel Multi-Domain Dataset Grounding LTL in Contextual Natural Language Via Provable Intermediate Logic

Paapa Kwesi Quansah, Pablo Rivas, Ernest Bonnah

Baylor University, Waco, Texas

The Specification Gap

$$G((t > 22{:}00 \wedge door\_open) \rightarrow lights\_on\ U\ \neg door\_open)$$

"If any exterior door opens after 10 p.m., the security lights should immediately turn on and stay on until the door is closed."

- Formal specifications are mathematically precise but opaque to domain experts.
- Natural language requirements are universally accessible but highly ambiguous and error-prone.
- Existing datasets attempting to bridge this gap are incredibly small (2–5k examples), confined to narrow robotic navigation commands, and produce rigid, template-like structures rather than contextual translations.

VERIFY: What We Built

200,916 triplets

13 Domains

Autonomous Vehicles, Medical Devices, Smart Grid Management, Build Pipelines and CI/CD, Networking/Distributed Systems, Home Automation, Security and Authentication, Financial/Transaction Systems, Web Services/APIs, Industrial Automation/Manufacturing, Robotics, Aerospace, Version Control and Code Reviews

3 Representation Layers

LTL → ITL → NL

>97% Validated

Manual expert review + LLM judge

Samples per Domain:  ∼16.7k each

LTL → ITL → NL

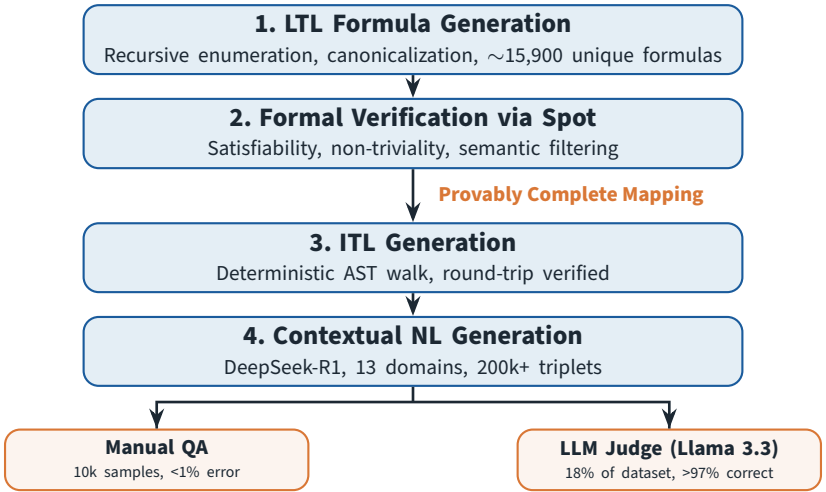
LTL: `G(system_ready → (check_a U check_b))`

ITL: `Always(IF system_ready THEN (check_a Until check_b))`

NL: *(Industrial Automation)* “It must always be the case that if the system reports ready, check A must remain valid until check B eventually becomes true.”

Completeness proof: Appendix C.

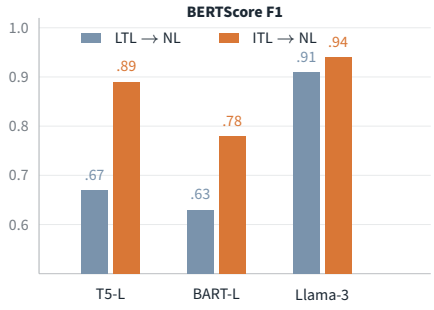
How We Built It



Comparison with Existing Datasets

Dataset	Scale	Domains	NL Type	ITL	Formal Verif.
Cherukuri et al.	∼12k	Abstract	Technical	–	Yes
Pan et al.	∼6.5k	Robotics	Commands	–	Yes
Wang et al.	∼2k	Robotics	Commands	–	Yes
Lang2LTL	∼2.1k	Navigation	Commands	–	Yes
NL2TL	∼28k	Lifted	Templates	–	Yes
Hahn et al.	∼100k	Hardware	Technical	–	Yes
VERIFY (ours)	**200k+**	**13 Diverse**	**Contextual**	✓	**✓ (Spot)**

Generating Natural Language from Logic



Fine-tuned LLMs successfully solve the forward generation task, achieving BERTScore F1 metrics above 0.91 for direct LTL $\rightarrow$ NL and 0.94 for ITL $\rightarrow$ NL translations.

The use of **ITL as an intermediate representation consistently improves generation quality** across all model architectures. T5-large demonstrates the most dramatic relative gain, jumping from 0.67 to 0.89 BERTScore F1 when conditioned on ITL.

Parsing Natural Language to Logic

Model	SemEq	EM	SynCorr
T5-large	22.3	2.8	66.1
Llama-3-8B	28.2	4.1	73.6
Mistral-7B	25.6	2.9	68.4
DeepSeek	**31.5**	**5.4**	**74.2**

Best semantic equivalence: 31.5%.

The reverse direction remains an open challenge.

Error Category	%
Incorrect Scope	41%
Op. Mismatch	28%
Prop. Atom Error	17%
Context Grounding	9%
Syntactic Malform.	5%

Syntactic errors are remarkably rare at 5%, demonstrating that models reliably learn the surface grammar of LTL.

The dominant failure mode is **operator scope**, where models produce parseable formulas with incorrect nesting, suggesting unconstrained autoregressive decoding struggles with rigid logical structures.

What the Analytical Experiments Show

Context Matters

Llama-3 NL → LTL (SemEq)

With Context	**28.2%**
Without	**7.7%**

Complexity Degrades

SemEq by AST Depth Bucket

Depth 1–4	**35.4%**
Depth 5–8	**28.1%**
Depth 9–12	**21.9%**
Depth 13+	**15.2%**

Cross-Domain Transfer

In-Domain vs. Out-of-Domain

	In	Out
Aerospace	27.5	19.2
Home	30.1	22.5
Finance	25.8	16.7

Summary

- VERIFY provides 200k+ formally verified LTL-ITL-NL triplets across 13 domains, establishing the largest and most diverse resource of its kind.
- The provably complete ITL intermediate representation yields consistent gains in natural language generation quality across all tested architectures.
- The NL $\rightarrow$ LTL direction remains far from solved, with the best model reaching 31.5% semantic equivalence and operator scope acting as the primary failure mode.
- Structured decoding methods and neurosymbolic architectures represent the most promising path forward for logic generation.

GitHub: github.com/sedislab/Verify
HuggingFace: huggingface.co/datasets/sedislab/VERIFY
Kaggle: kaggle.com/datasets/sedislab/verify