Evaluating Autoformalization Robustness via Semantically Similar Paraphrasing

Hayden Moore[1]*, **Asfahan Shah**[1]*

[1]Department of Computer Science and Engineering, The Pennsylvania State University
University Park, PA 16802 USA
{hmm5731, aks7824}@psu.edu

Abstract

Large Language Models (LLMs) have recently emerged as powerful tools for autoformalization. Despite their impressive performance, these models can still struggle to produce grounded and verifiable formalizations. Recent work in text-to-SQL, has revealed that LLMs can be sensitive to paraphrased natural language (NL) inputs, even when high degrees of semantic fidelity are preserved (Safarzadeh, Oroojlooyjadid, and Roth 2025). In this paper, we investigate this claim in the autoformalization domain. Specifically, we evaluate the robustness of LLMs generating formal proofs with semantically similar paraphrased NL statements by measuring semantic and compilation validity. Using the formal benchmarks MiniF2F (Zheng, Han, and Polu 2021) and Lean 4 version of ProofNet (Xin et al. 2024), and two modern LLMs, we generate paraphrased natural language statements and cross-evaluate these statements across both models. The results of this paper reveal performance variability across paraphrased inputs, demonstrating that minor shifts in NL statements can significantly impact model outputs.

Introduction

Recent research has shown progress in *autoformalization*, defined as the translation of natural language (NL) mathematical statements into formal proofs. By combining modern large language models (LLMs) with symbolic theorem provers such as Isabelle (Nipkow, Paulson, and Wenzel 2002) and Lean (de Moura et al. 2015), there is hope for making formal mathematical verifications accessible to anyone who can articulate a problem in NL. Closing the gap between a novice understanding of math and formal proofs, changing how knowledge could be authored and verified.

Despite these recent advances, using LLMs for autoformalization remains far from a verifiable and trusted system that is deployable in practice. Currently, these systems often generate formal statements that could be syntactically valid but remain unverifiable, logically inconsistent, or completely wrong. Things like unintended insertions, omissions, misinterpretations, or hallucinations demonstrate not only errors in translation but deeper failures of foundational model integrity. This is especially important in a domain like mathematics, where correctness is an absolute requirement, and where one small error can undermine the reliability of the entire generated proof chain.

These errors can erode trust and limit adoption into real-world applications. To adopt LLM autoformalization into a serious workflow, we must first validate that these systems produce correct outputs and be able to explain why its correct. Without this necessary interpretability and explainability, the future of machine assisted formal reasoning risks becoming a black-box system, which experts typically hesitate to adopt. Only by embedding explainability and interpretability at the core of these systems and benchmarks can we help ensure that the future of autoformalization enhances without obscuring our understanding of the system.

Related Work

Recent research has reported that linguistic variations can lead to significant performance degradation in the text-to-SQL domain, stating *"This problem underscores the sensitivity of current models and the need for more resilient solutions"* (Safarzadeh, Oroojlooyjadid, and Roth 2025). This begs the question of whether such sensitivity extends to other domains that also rely on LLMs as their core component.

In the domain of mathematics, the concept of autoformalization using LLMs was first demonstrated by (Wu et al. 2022), where LLMs were used to formalize natural-language mathematical statements into Isabelle/HOL. Since then, impressive progress has been made in advancing autoformalization techniques. Building on these advances (Yang et al. 2023) developed LeanDojo, which provides unified infrastructure and datasets for training and evaluating LLMs on formalization tasks in Lean. Similarly (Jiang et al. 2022) introduced Thor, demonstrating how LLMs can be integrated with automated theorem provers to solve formal proofs in Isabelle.

The MiniF2F benchmark (Zheng, Han, and Polu 2021) introduced a dataset for Olympiad-level formal reasoning enabling evaluation of LLM performance in proof synthesis for formal languages such as Isabelle (Nipkow, Paulson, and Wenzel 2002). Additionally, ProofNet (Azerbayev et al. 2023) expanded the scale and diversity of mathematical problems, bridging undergraduate-level mathematics with formal proofs to evaluate LLM's performance in proof syn-

*These authors contributed equally.

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

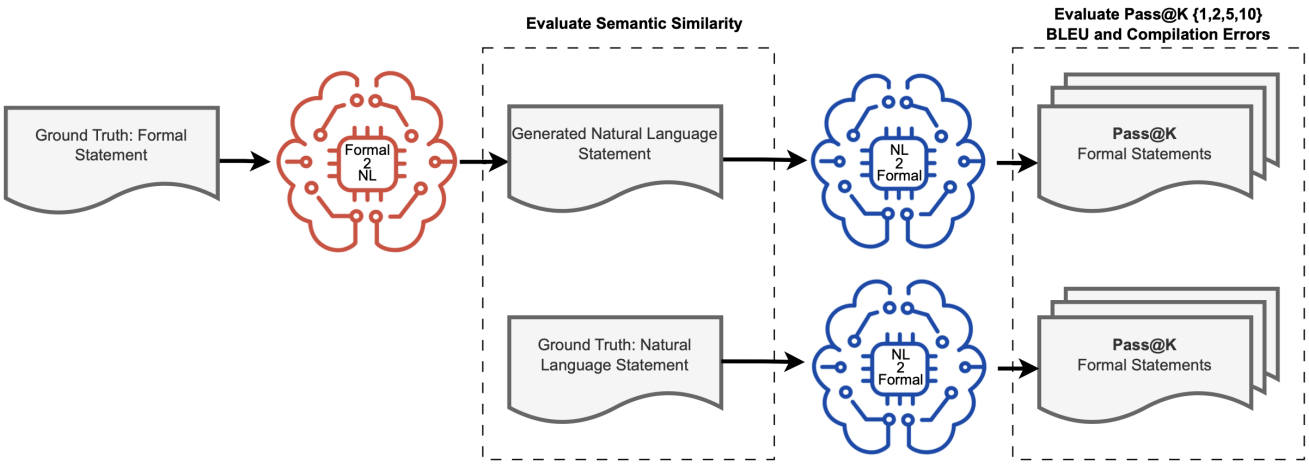


Figure 1: Overview of the autoformalization robustness evaluation pipeline for MiniF2F(Isabelle/HOL) and ProofNet(Lean 4). Each formal system undergoes both forward (Formal→NL) and reverse (NL→Formal) paraphrasing stages, using both GPT-4o-mini (OpenAI 2024) and Claude-3.7-sonnet (Anthropic 2025), with consistency evaluated via similarity metrics and Pass@K accuracy.

thesis for Lean 3 code (de Moura et al. 2015). (Xin et al. 2024) later introduced a Lean 4 version of ProofNet.

Preliminaries

BLEU Evaluation. We adopt the BLEU formalization used in Zheng, Han, and Polu (2021), this metric was originally proposed by Papineni et al. (2002). Given a reference sequence r and a candidate sequence c, the BLEU score is defined as:

$$\mathrm{BLEU} = \mathrm{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right), \tag{1}$$

where p_n denotes the modified n-gram precision up to order N, w_n is the weight (typically $w_n = \frac{1}{N}$), and BP is the brevity penalty:

$$\mathrm{BP} = \begin{cases} 1, & \text{if } c_{\text{len}} > r_{\text{len}}, \\ \exp\left(1 - \frac{r_{\text{len}}}{c_{\text{len}}}\right), & \text{otherwise.} \end{cases} \tag{2}$$

We follow the MiniF2F and ProofNet evaluation protocol by computing sentence-level BLEU with smoothing (method 1 from NLTK) to account for sparse n-gram overlap in short formal proofs.

Lexical Diversity. To quantify variation in word usage across paraphrased and reference statements, we compute *lexical diversity* as the type–token ratio (TTR) (Tweedie and Baayen 1998):

$$\mathrm{TTR} = \frac{|V|}{|W|}, \tag{3}$$

where $|V|$ is the number of unique tokens and $|W|$ is the total number of tokens in the natural language statement.

Cosine Similarity. To measure semantic equivalence between the paraphrased and reference statements, we compute the cosine similarity of their sentence embeddings:

$$\mathrm{Sim}_{\cos}(r, c) = \frac{E(r) \cdot E(c)}{\|E(r)\| \|E(c)\|}, \tag{4}$$

where $E(\cdot)$ denotes the SBERT (Reimers and Gurevych 2019) embedding function.

Methodology

For our methodology we follow a two staged process to evaluate how well autoformalization performs when introduced with paraphrased inputs (Figure 1). **Each stage is context independent, meaning that all LLM requests do not have context to previous requests.**

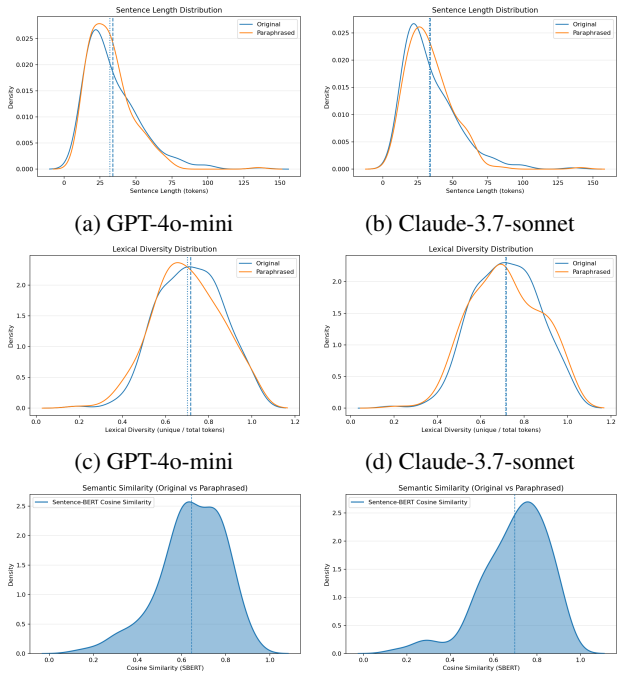
The first stage is where we generate the paraphrased NL statements using two modern LLMs, GPT-4o-mini (OpenAI 2024) and Claude-3.7-sonnet (Anthropic 2025). Each formal statement from MiniF2F and ProofNet (Lean 4), is passed as input to the LLMs, prompting them to translate the formal proof into a NL statement that accurately captures the logic of the proof (prompts defined in the Appendix). We also perform a semantic similarity analysis on these paraphrased NL statements and the corresponding ground truth NL statement, which is defined in the results section of this paper. This stage establishes the semantic validity of the paraphrased NL statements before we evaluate the translation performance for NL→Formal.

The second stage is where we perform a Pass@K cross-evaluation of GPT and Claude paraphrased NL statements. Testing each models ability and sensitivity to handling variations of the original NL statement that are still semantically similar. We pass GPT/Claude paraphrased NL statements to both GPT/Claude models and evaluate their performance on

both BLEU accuracy and compilation accuracy. BLEU accuracy is defined in the previous preliminaries section. Compilation accuracy is defined by the successful execution of the generated proof code (Isabelle/Lean 4) with the respective compilers.

Results

Semantic Similarity Analysis



(a) GPT-4o-mini (b) Claude-3.7-sonnet

(c) GPT-4o-mini (d) Claude-3.7-sonnet

(e) GPT-4o-mini (f) Claude-3.7-sonnet

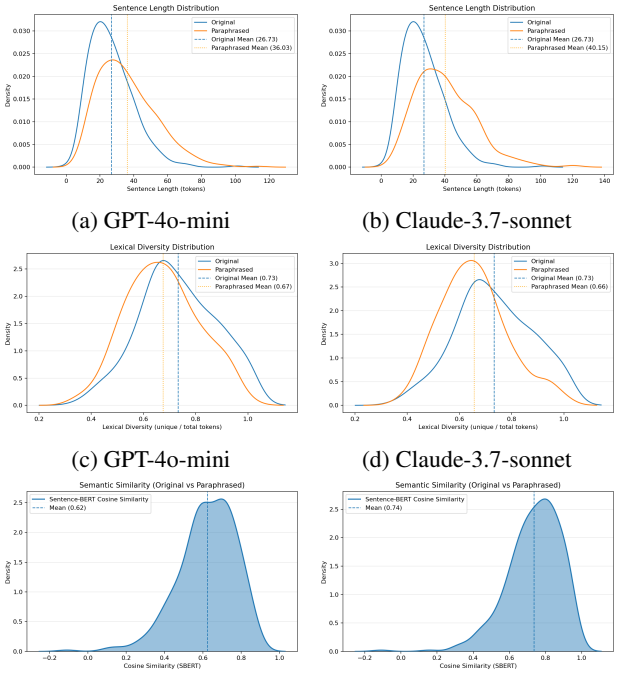
Figure 2: **MiniF2F (Isabelle)**: Panels show sentence length, lexical diversity, and SBERT semantic similarity for GPT-4o-mini and Claude-3.7 paraphrasings.

We first create a baseline of the semantic similarity between the paraphrased NL statements and their corresponding ground truth formal NL statements (Figure 2).

Our results with **MiniF2F (Isabelle)** using both GPT-4o-mini and Claude-3.7-sonnet for paraphrasing, we observe consistent and high cosine similarity distributions relative to the ground truth NL statements (64-72%). Indicating that the paraphrasing step largely preserves semantic meaning even across the diversity of the benchmark.

For **ProofNet (Lean 4)** (Figure 3), when using both GPT-4o-mini and Claude-3.7-sonnet for paraphrasing, we also observe high cosine similarity distributions relative to the ground truth NL statements (62-78%). Indicating once again that the paraphrasing step preserves semantic meaning. However, we do observe a more loose clustering around the centroid for lexical diversity and sentence length, indicating that the Formal→NL translation for Lean 4 seems to introduce more lexical diversity.

These results help us confirm that the paraphrasing stage maintains a high semantic equivalence across both models tested while still providing linguistic variations to the statement. This helps ensure that our subsequent variations in formalizations stems less from semantic meaning drift but more from the LLMs sensitivity to the linguistic differences in the paraphrased statements. In other words, the LLMs are given semantically equivalent paraphrased statements and we test if the autoformalization step remains invariant.



(a) GPT-4o-mini (b) Claude-3.7-sonnet

(c) GPT-4o-mini (d) Claude-3.7-sonnet

(e) GPT-4o-mini (f) Claude-3.7-sonnet

Figure 3: **ProofNet (Lean 4)**: Panels show sentence length, lexical diversity, and SBERT semantic similarity for GPT-4o-mini and Claude-3.7 paraphrasings.

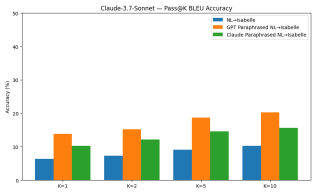
Pass@K Evaluation

The Pass@K results in Table 1 reveal clear cross-model dependencies between the paraphraser and the formalization model. While both models exhibit improved average BLEU and compilation performance when using paraphrased inputs, the magnitude and direction of these improvements appear to be sensitive to the paraphrase source.

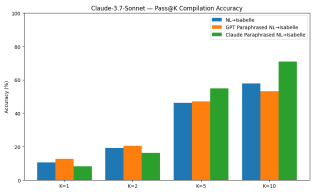
MiniF2F: Isabelle/HOL Our results from the MiniF2F benchmark experiments reveal clear variability in performance across paraphrased inputs (Figure 4). First, when GPT-4o-mini is used as the formalization model, GPT-paraphrased inputs lead to higher BLEU for all K and compilation scores for Pass@$\{1,2,5\}$ compared to the ground-truth and Claude-paraphrased inputs. When the GPT paraphrases are evaluated by Claude-3.7-sonnet, the semantic fidelity (BLEU) remains strong, while compilation accuracy is stronger at Pass@$\{1,2\}$ but less pronounced at Pass@$\{5,10\}$ and having the lowest Pass@10 score for the Claude formalization experiments. However, if we just look at the BLEU scores for the GPT-paraphrased NL statements,

we see across the board, for all K, they are the highest.

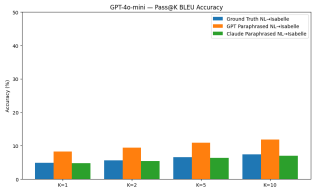
Conversely, Claude-paraphrased inputs exhibit stronger generalization effects for compilation accuracy. The model achieves 70.9% Pass@10 compilation, which is the highest score across all configurations. Interestingly, BLEU scores for Claude paraphrases lag slightly behind GPT paraphrases, implying that while Claude's phrasing improves logical structure and compilation validity, it sometimes departs further from the distributions of the reference statement. When evaluated by GPT-4o-mini, these same paraphrases retain moderate BLEU similarity and display good compilation accuracy at Pass@{5,10}.



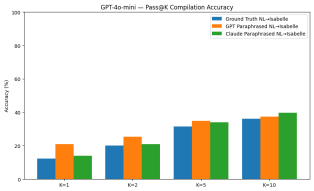
(a) Claude-3.7-Sonnet, BLEU



(b) Claude-3.7-Sonnet, Comp.



(c) GPT-4o-mini, BLEU



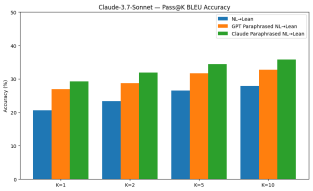
(d) GPT-4o-mini, Comp.

Figure 4: **MiniF2F (Isabelle)** autoformalization cross-evaluation results. Pass@K accuracy is reported for $K \in \{1, 2, 5, 10\}$ across semantic (BLEU) and syntactic (Compilation) metrics. Results are shown for both formalization models: Claude-3.7-Sonnet and GPT-4o-mini.

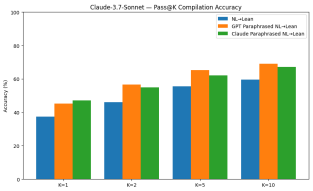
ProofNet: Lean 4 Our results from the ProofNet (Lean 4) benchmark experiments also reveal variability in performance across paraphrased inputs (Figure 5). First, the magnitude of both BLEU and compilation changes is more pronounced than with our Isabelle experiments. When GPT-4o-mini is the formalization model, both paraphrased inputs improve BLEU over the ground-truth NL statements across all K. However, compilation accuracy for GPT-4o-mini declines slightly under paraphrasing, indicating that subtle syntactic shifts may have been introduced that affect executability in Lean 4's strict type system. These findings highlight that paraphrasing can sometimes succeed with meaning preservation but does not always lead to syntactic validity.

In contrast, Claude-3.7-Sonnet demonstrates slightly stronger robustness and transferability across paraphrased inputs. BLEU steadily increases from 20.58% (ground-truth) to 29.25% under Claude paraphrasing, and compilation accuracy rises sharply from 37.46% to 47.16% at Pass@1, with gains persisting through Pass@10 (59.56 $\rightarrow$ 67.11). GPT paraphrased inputs also show notable improvements.

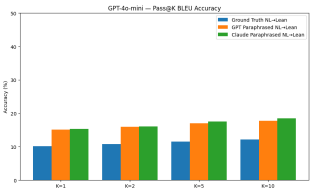
Overall, the Lean 4 results reinforce the trend observed in Isabelle: paraphrasing can meaningfully shift model performance, and robustness to linguistic variation remains an open challenge for current autoformalization systems.



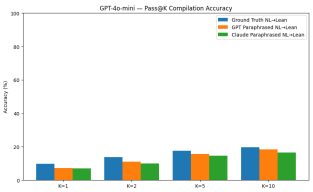
(a) Claude-3.7-Sonnet, BLEU



(b) Claude-3.7-Sonnet, Comp.



(c) GPT-4o-mini, BLEU



(d) GPT-4o-mini, Comp.

Figure 5: **ProofNet (Lean 4)** autoformalization cross-evaluation results. Pass@K accuracy is reported for $K \in \{1, 2, 5, 10\}$ across semantic (BLEU) and syntactic (Compilation) metrics. Results are shown for both formalization models: Claude-3.7-Sonnet and GPT-4o-mini.

Conclusion

Our results demonstrate that even when paraphrased statements maintain a high semantic similarity to their ground truth counterparts, current LLM-based autoformalization systems exhibit sensitivity in both semantic fidelity and syntactic validity. These results align with and extend the generalizability of recent findings in the text-to-SQL domain (Safarzadeh, Oroojlooyjadid, and Roth 2025), which similarly report that LLMs are highly sensitive to minor linguistic perturbations while preserving semantic meaning. Thus, there is a need for more robust autofomalization pipelines that can mitigate this sensitivity to minor linguistic perturbations, ensuring more consistent results.

Limitations

This paper performs a cross-evaluation of paraphrased NL statements for two models, future research should expand this to more models to see if the sensitivity to paraphrased inputs persists, or if some models are more robust. Additionally, we perform an evaluation on semantically similar paraphrased NL statements but this work does not explore the results for low semantic similarity, ambiguous, or inconsistent proof requests. Finally, this work does not perform a systematic evaluation of the semantic or compilation error categories, which could provide deeper insights into the sensitivity of paraphrased inputs.

Table 1: Pass@K Evaluation for Autoformalization on ProofNet(Lean 4) and MiniF2F(Isabelle/HOL). We report both Pass@K BLEU Accuracy (semantic fidelity) and Pass@K Compilation Accuracy (syntactic validity) across varying K values. Each section corresponds to the LLM used for formalization.

Model / Setting	**Pass@K BLEU Accuracy (%)**				**Pass@K Compilation Accuracy (%)**			
	K=1	K=2	K=5	K=10	K=1	K=2	K=5	K=10
GPT-4o-mini (Formalization Model)								
Ground Truth NL→Isabelle	4.84	5.57	6.54	7.41	12.30	20.08	31.56	36.07
GPT Paraphrased NL→Isabelle	8.31	9.39	10.92	11.87	20.90	25.41	34.84	37.30
Claude Paraphrased NL→Isabelle	4.77	5.36	6.32	7.00	13.93	20.90	34.02	39.75
Ground Truth NL→Lean 4	10.12	10.78	11.50	12.21	9.70	13.74	17.52	19.67
GPT Paraphrased NL→Lean 4	15.10	16.00	16.97	17.75	7.27	11.05	15.63	18.32
Claude Paraphrased NL→Lean 4	15.36	16.06	17.50	18.44	7.00	9.97	14.55	16.44
Claude-3.7-Sonnet (Formalization Model)								
NL→Isabelle	6.35	7.30	9.06	10.29	10.66	19.26	46.31	57.79
GPT Paraphrased NL→Isabelle	13.90	15.24	18.73	20.25	12.70	20.49	47.13	53.28
Claude Paraphrased NL→Isabelle	10.25	12.12	14.56	15.68	8.20	16.39	54.92	70.90
NL→Lean 4	20.58	23.37	26.51	27.83	37.46	46.09	55.52	59.56
GPT Paraphrased NL→Lean 4	26.98	28.76	31.63	32.70	45.28	56.60	65.22	69.00
Claude Paraphrased NL→Lean 4	29.25	31.90	34.46	35.78	47.16	54.98	61.99	67.11

References

Anthropic. 2025. Claude 3.7 Sonnet: Hybrid Reasoning Model from Anthropic. https://www.anthropic.com/news/claude-3-7-sonnet. Accessed: 2025-10-31.

Azerbayev, Z.; Piotrowski, B.; Schoelkopf, H.; Ayers, E. W.; Radev, D.; and Avigad, J. 2023. ProofNet: Autoformalizing and Formally Proving Undergraduate-Level Mathematics. *arXiv preprint arXiv:2302.12433*.

de Moura, L.; Kong, S.; Avigad, J.; Van Doorn, F.; and von Raumer, J. 2015. The Lean Theorem Prover (System Description). In *Automated Deduction – CADE-25*, volume 9195 of *Lecture Notes in Computer Science*, 378–388. Springer.

Jiang, A. Q.; Li, W.; Tworkowski, S.; Czechowski, K.; Odrzygóźdź, T.; Miłoś, P.; Wu, Y.; and Jamnik, M. 2022. Thor: Wielding Hammers to Integrate Language Models and Automated Theorem Provers. arXiv:2205.10893.

Nipkow, T.; Paulson, L. C.; and Wenzel, M. 2002. Isabelle/HOL — A Proof Assistant for Higher-Order Logic. In *Theorem Proving in Higher Order Logics (TPHOLs 2002)*, volume 2283 of *Lecture Notes in Computer Science*, 1–16. Springer.

OpenAI. 2024. GPT-4o mini: Advancing Cost-Efficient Intelligence. https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/. Accessed: 2025-10-31.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. BLEU: a method for automatic evaluation of machine translation. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, 311–318.

Reimers, N.; and Gurevych, I. 2019. Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP) and the 9th International Joint Conference on Natural Language Processing (IJCNLP)*, 3982–3992. Association for Computational Linguistics.

Safarzadeh, M.; Oroojlooyjadid, A.; and Roth, D. 2025. Evaluating NL2SQL via SQL2NL. *arXiv preprint arXiv:2509.04657v1*. Affiliation: Oracle AI.

Tweedie, F. J.; and Baayen, R. H. 1998. *Measuring Lexical Diversity*, volume 32. Springer.

Wu, Y.; Jiang, A. Q.; Li, W.; Rabe, M. N.; Staats, C.; Jamnik, M.; and Szegedy, C. 2022. Autoformalization with Large Language Models. arXiv:2205.12615.

Xin, H.; Ren, Z. Z.; Song, J.; Shao, Z.; Zhao, W.; Wang, H.; Liu, B.; Zhang, L.; Lu, X.; Du, Q.; Gao, W.; Zhu, Q.; Yang, D.; Gou, Z.; Wu, Z. F.; Luo, F.; and Ruan, C. 2024. DeepSeek-Prover-V1.5: Harnessing Proof Assistant Feedback for Reinforcement Learning and Monte-Carlo Tree Search.

Yang, K.; Swope, A. M.; Gu, A.; Chalamala, R.; Song, P.; Yu, S.; Godil, S.; Prenger, R.; and Anandkumar, A. 2023. LeanDojo: Theorem Proving with Retrieval-Augmented Language Models. arXiv:2306.15626.

Zheng, K.; Han, J. M.; and Polu, S. 2021. MiniF2F: a cross-system benchmark for formal Olympiad-level mathematics. *arXiv preprint arXiv:2109.00110*.

Appendix

Prompts for Autoformalization Evaluation

We provide the exact prompts used for our autoformalization experiments. These prompts were used with

GPT-4o-mini/Claude-3.7-sonnet for both directions: (1) Natural Language to Isabelle/ Lean 4 Formalization (NL→Formal), and (2) Paraphrased evaluation (Formal→NL→Formal). All generations were executed with temperature = 0.0, top_p = 1.0, max tokens = 500 (for single generation), max tokens = 5000 (for multiple generations), no context between tasks in roundtrip.

Prompt 1: Ground-Truth NL → Formal (Isabelle)

System: You are a helpful assistant that translates between formal logic and natural language.

User: Translate the following Natural Language statement into a [Isabelle] Formal Statement that conveys the exact same logical meaning. Generate only the [Isabelle] Formal Statement, without any additional commentary or explanation.

Natural Language Statement: [natural_s]

Formal Statement:

Prompt 2: Ground-Truth Formal → NL (Isabelle/Lean 4)

System: You are a helpful assistant that translates between formal logic and natural language.

User: Translate the following [Isabelle/Lean 4] statement into a clear natural language question that conveys the exact same logical meaning. Generate only the natural language question written in LaTex, without any additional commentary or explanation.

Formal Statement: [formal_s]

Natural Language Statement:

Prompt 3: NL → Formal, Multiple Generation (Isabelle)

System: You are a helpful assistant that translates between formal logic and natural language.

User: Translate the following natural language statement into 10 clear [Isabelle] statements that conveys the exact same logical meaning. Generate a numbered list of 10 unique [Isabelle] statements, without any additional commentary or explanation.

Natural Language Statement: [natural_s]

Formal Statement: 1.

Prompt 4: Ground-Truth NL → Formal (Lean 4)

System: You are a helpful assistant that translates between formal logic and natural language.

User: Translate the following Natural Language (or LaTeX) statement into a clear, valid [Lean 4] theorem that convey the same logical meaning without any additional commentary or explanation. Requirements:

1. Output (Translated [Lean 4] statement) must be a string assigned to the output field message.
2. Translated statement must:
   - Begin with 'theorem'
   - Be a self-contained [Lean 4] statement
   - Do not include import lines
   - Not include 'by', 'sorry', or 'import'
   - Encode any necessary assumptions in variable names or hypotheses
3. Make reasonable assumptions if the natural language statement is underspecified.
4. Do not add any commentary or explanations.

[natural_s]

Prompt 5: NL → Formal, Multiple Generation (Lean 4)

System: You are a helpful assistant that translates between formal logic and natural language.

User: Translate the following Natural Language (or LaTeX) statement into exactly 10 clear, valid [Lean 4] theorems that convey the same logical meaning without any additional commentary or explanation. Requirements:

1. Output must be a Python list assigned to the output field message.
2. The list must contain exactly 10 string elements (each element is a valid [Lean 4] translation), no more, no less.
3. Each element must:
   - Begin with 'theorem'
   - Be a single, self-contained [Lean 4] statement
   - Do not include import lines
   - Not include 'by', 'sorry', or 'import'
   - Encode any necessary assumptions in variable names or hypotheses
4. Make reasonable assumptions if the natural language statement is underspecified.
5. Do not add any commentary or explanations.

[[natural_s]