Assessing the Alignment of FOL Closeness Metrics with Human Judgement

Anonymous ACL submission

Abstract

The recent successful paradigm of solving logical reasoning problems with tool-augmented large language models (LLMs) leverages translation of natural language statements into First-Order Logic (FOL) and external theorem provers. However, the correctness of FOL statements, comprising operators and text predicates, often goes unverified due to the lack of a reliable evaluation metric for comparing generated and ground-truth FOLs. In this paper, we present a comprehensive study of sensitivity of existing metrics and their alignment with human judgement on FOL evaluation. Using ground-truth FOLs, we carefully designed various perturbations on the ground-truth to assess metric sensitivity. We sample FOL translation candidates for natural language statements and measure the ranking alignment between automatic metrics and human annotators. Our empirical findings highlight oversensitivity in the n-gram metric BLEU for text perturbations, the semantic graph metric Smatch++ for structural perturbations, and FOL metric for operator perturbation. We also observe a closer alignment between BertScore and human judgement. Additionally, we show that combining metrics enhances both alignment and sensitivity compared to using individual metrics. [1]

1 Introduction

Large language models (LLMs) have advanced natural language reasoning, but logical and mathematical reasoning have long relied on formal, structured languages for proving deductions and theorems, a process that predates deep neural networks (Quiñonero-Candela et al., 2006). This approach remains relevant today, especially for reasoning tasks that can be solved using formal statements. In case of first-order logic (FOL), LLM generations are used as intermediate steps and subsequently passed to theorem provers to solve the problem (Pan et al., 2023; Ye et al., 2024; Olausson et al., 2023). Compared to the Chain-of-Thought (CoT) approach (Wei et al., 2022), where the model first reasons and then solves, FOL generation demonstrated superior reliability by offloading the reasoning task to an external tool. Translating natural language (NL) into FOL enhanced the overall rigor of the process.

Generating FOL from NL is a challenging task that tests the ability of LLMs to accurately interpret and convert informal language into a formal, structured token sequence. The lack of ground truth for FOL generations complicates direct verification. Yang et al. (2024) addressed this challenge by developing a system specifically for FOL generation, incorporating an operator-based evaluator to rate the outputs. This evaluation is combined with BLEU score, using a threshold as a metric in a reward model. However, the reliance on thresholds complicates the interpretation of translation quality. Manually assessing formal logic generations is labor-intensive and has received relatively less attention compared to traditional text translation metrics. In this work, we analyze existing natural language translation, tree, and graph evaluation metrics, focusing on those that offer strong sentence-level comparisons.

We establish a framework to systematically introduce perturbations and analyze the existing metrics in the presence of these anomalies in formal language, specifically first-order logic. To further assess the robustness of these metrics, we sample FOLs from NL statements in FOLIO dataset (Han et al., 2022) using an LLM and rank them against ground truth values. The ranking is conducted using established metrics, LLM-based evaluators, human annotations, and combinations of metrics. Our findings provide valuable insights into the effectiveness of current metrics and their applicability to symbolic generation tasks.

[1] Our code is available at https://anonymous.4open.science/r/AlignmentFOL-CBF0/

2 Closeness Metrics

Evaluation scores in natural language generation, such as BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004), perform n-gram matching between reference text and candidate outputs. METEOR (Banerjee and Lavie, 2005), while also based on n-gram overlap, incorporates additional factors known to result in improved correlation with human judgments. BERTScore (Zhang et al., 2019) leverages contextual embeddings generated by a pre-trained BERT (Devlin et al., 2019) to compute cosine similarity between sentences.

In contrast, logical equivalence (Yang et al., 2024) evaluates FOL translations by comparing the truth values of formal statements, abstracting away from their textual semantics. Another relevant domain is Abstract Meaning Representation (AMR) graph metrics, which compare the structural similarity of semantic graphs. Given the structured nature of FOL statements, we leverage Smatch++ (Opitz, 2023), which incorporates preprocessing, alignment, and sub-graph scoring. These metrics capture different dimensions of divergence between ground truth and translations: traditional metrics focus on surface-level and semantic discrepancies, while formal evaluation methods assess deeper logical consistency. We present results demonstrating how a representative set of these metrics respond to variations in logical constructs within formal language translations.

3 Evaluation Framework

3.1 Perturbation Evaluations

The effect of perturbations measures the *sensitivity* of the metrics by assessing how small changes or variations in the FOL statements impact the metric scores. Based on the ground-truth of the FOLIO dataset (Han et al., 2022), we utilize nine operators to construct a formal logic framework. Perturbation variations are applied to assess the impact on metric scores. To evaluate the performance of these metrics, we first conduct a self-matching experiment on the statements and normalize the results based on the variations observed in this process (Table 2). The perturbation strategies are as following:

Quantifier: In this perturbation, we swap the quantifiers $\forall$ and $\exists$ where applicable. For example, the formula $\forall x(W(x, C) \rightarrow A(x, C))$ becomes $\exists x(W(x, C) \rightarrow A(x, C))$. This subtle change allows us to isolate the effect of quantifiers on the similarity metrics, demonstrating that misidentifications in quantifier use can be detected by these metrics.

Negation: This perturbation measures the impact of negation on the metrics. We either remove the negation of predicates, if present, or add it when absent. For example, $\forall x(\neg W(x, C) \rightarrow A(x, C))$ changes to $\forall x(W(x, C) \rightarrow \neg A(x, C))$. This modification tests the metrics' ability to correctly identify the placement of negations, maintaining fidelity to the underlying logic.

And/Or and Or/Xor: This perturbation involves a simple swap of logical operators, such as (And, Or) and (Or, Xor). Given that translations by the LLMs may confuse these operators, it is important to assess how such changes are reflected in the metrics.

Operator: This perturbation focuses on the role of operators in influencing similarity scores. All logical operators are removed, and any multiple predicates are connected by a disjunction ($\vee$) to preserve the structure. For instance, $\forall x(\neg W(x, C) \rightarrow A(x, C))$ becomes $W(x, C) \vee A(x, C)$.

Predicate: This perturbation modifies all predicates 'P' containing a negation by converting them to their 'NotP' form. It tests the metrics' ability to detect variations in both negation and semantics. For example, $(\neg \text{WantToBeAddictedTo}(\text{caffeine}))$ is transformed to $(\text{NotWantToBeAddictedTo}(\text{caffeine}))$.

Variable: This perturbation is designed to examine the metrics' ability to handle semantic changes. All text values are replaced with generic variables and compared with the ground truth. For example, $\forall x(\neg \text{WantToBeAddictedTo}(x, \text{caffeine}) \rightarrow \text{AwareThatDrug}(x, \text{caffeine}))$ becomes $\forall x(\neg A(x, C) \rightarrow B(x, C))$. [2] [3]

3.2 Sample Evaluations

Measuring the sample correctness with respect to the ground truth allows for an assessment of

[2] All previous examples, except for 'Variable' and 'Predicate', have been shortened for space, but expanded forms are used in the experiments

[3] The perturbations in the "Operator" and "Variable" sections introduce free variables that can be confusing to interpret because of the lack of quantification. hese errors pass through the tool without triggering any issues, making them a common occurrence in FOL generations by LLMs. Identifying this problem highlights a significant gap in the reliability of LLM-generated FOL translations.

alignment between different types of rankers. We randomly sampled a small set of data from FOLIO dataset and implemented a sampling process in which `gpt-4o` (Achiam et al., 2023) was zero-shot prompted to generate three samples of FOLs for each text input. For each data point, consisting of a natural language text and its corresponding FOL label, denoted as $\{NL, FOL\}$, the NL was provided to `gpt-4o` (see Appendix B for prompt detail) to generate three samples: $\{FOL_1, FOL_2, FOL_3\}$. In many cases, the LLM produced a correct FOL_1. To introduce randomness, we shuffled the order of the samples before passing them to the metrics. The shuffled FOL samples, $\{\hat{FOL}_1, \hat{FOL}_2, \hat{FOL}_3\}$, were then evaluated by various metrics, with a score produced for each comparison. In instances when two or all three comparisons resulted in tied scores, the ranks were adjusted to be equal. For example, if the first two FOLs were the same and the third was different, with initial ranks of [1, 2, 3], we adjusted them to [1, 1, 3].

To compare the rankings generated by these metrics, we conducted a human evaluation. We enlisted three annotators with at least a Master's degree in CS or AI to rank the similarity between the ground truth FOL and the generated samples. The instructions provided to the experts were kept open-ended, offering only a basic overview of the logic and ranking criteria to avoid inducing bias (See Appendix C). Although the instructions suggested ranking randomly in case of a tie, we deduplicated the values and assigned the same rank to the matching FOLs, as described in the previous passage. We also used `gpt-4o` and `o1-preview` LLMs to rank the samples, allowing us to form a broader perspective on the results (see Appendix D for prompt details).

4 Experiments

Data Preparation. We use the training set of the FOLIO dataset for our experiments.[4] The operators are extracted from each record, and a unique set of operators is selected. Since our focus is on individual FOL statements, we decompose the records into single data points. To manage generation costs, we extract 102 records, ensuring a diverse combination of operators. Upon review, we observe that the number of operators in the records ranges from 0 to 7, with 0 representing a standalone predicate. The detailed data statistics are provided in Appendix A.

[4]We limit the data to one type and choose the training set to ensure diversity.

	Match $=$	Quantifier $\rightarrow$	Negation $\rightarrow$	AndOr $\rightarrow$	OrXor $\rightarrow$	Operator $\rightarrow$	Predicate $=$	Variable $\rightarrow$
Data (%)	100	70	99	59	32	98	22	100

Table 1: Percentage of perturbations applied to 102 records. $\downarrow$ indicates preference for lower values, $=$ requires the values to remain the same after perturbation.

4.1 Evaluation Preparation

Perturbation. The perturbations are evaluated using six metrics: BLEU (BL), ROUGE (RO), METEOR (ME), Logical Equivalence (LE), BERTScore (BS), and Smatch++ (SP). Following the method outlined by Yang et al. (2024), we first convert the FOL statements into a parsable format for each metric. For LE, an additional syntax check is conducted to ensure the truth value of the FOL statement is valid before comparison. Due to the nature of the perturbations, they are applied only to relevant records. For example, quantifier perturbation is possible only if the statement contains a quantifier. The percentage of data used for each perturbation is provided in Table 1.

LLM Samples. We use `gpt-4o` with temperature 0.8 to generate three samples for each input (Appendix B). Samples that do not adhere to correct syntax or where all three generations are identical are discarded, reducing the dataset to 87 records. The rankings are based on a scale of [1, 2, 3], where 1 represents the best match and 3 the least match to the ground-truth. The LLM evaluation is conducted using gpt4o and o1-preview, where the model is prompted to rank the three samples in the same format as the human evaluators (Appendix C and Appendix D).

Pairwise Ranking. A pairwise comparison is performed between the three human annotations to determine the final rankings. For each pair of annotations, we compare their relative rankings and use these comparisons to establish the overall order. This approach ensures that the final ranking is derived by consistently evaluating each annotation against the others in a pairwise manner. Additionally, perturbation and sample evaluations are conducted using a combination of metrics to assess the effect of metric mixtures. To do this, we calculate

	Pertb	BL	LE	RO	ME	BS	SP
	Match =	1.00	1.00	1.00	1.00	1.00	1.00
Operator	Quantifier ↓	0.96	1.00	0.96	0.96	1.00	0.99
	Negation ↓	0.69	0.73	0.93	0.85	0.97	**0.37**
	AndOr ↓	0.88	**0.72**	0.95	0.96	1.00	0.96
	OrXor ↓	0.95	**0.92**	0.98	0.98	1.00	0.99
Text	Operator ↓	**0.20**	0.62	0.53	0.42	0.89	0.44
	Predicate =	0.94	0.93	0.97	0.98	**1.00**	0.92
	Variable ↓	**0.28**	1.00	0.74	0.68	0.92	0.76

Table 2: Evaluation Metrics Result on the comparison between ground-truth and perturbations performed under each corresponding row. The **bold** values indicate the best-performing metric score for each perturbation.

Quant↓	BL	LE	RO	ME	BS	SP
BL	0.96	0.94	0.96	0.96	0.97	**0.93**
LE	0.94	1.00	0.95	0.94	0.96	**0.92**
RO	0.96	0.95	0.96	0.96	0.98	**0.94**
ME	0.96	0.94	0.96	0.96	0.97	**0.93**
BS	0.97	0.96	0.98	0.97	1.00	**0.95**
SP	0.93	**0.92**	0.94	0.93	0.95	0.99

Table 3: The values along the diagonal (highlighted) represent individual scores for quantifier perturbation, while the off-diagonal values correspond to combined evaluators. 'BL-BL' indicates the use of the BLEU score metric alone, whereas 'BL-RO' represents the combination of BLEU and ROUGE.

the scores for each FOL and compute the average score for each sentence. These averages are then processed to obtain the final value.

5 Results and Discussion

We present results from the two variations.

Perturbation Analysis. From Table 2, we observe that quantifier perturbations have minimal impact overall. However, when metrics are combined (Table 3), Smatch++ proves to be a more sensitive metric for detecting changes in quantifiers. This trend is also evident in other metrics, where the use of combined metrics results in more distinct and consistent scores (discussed in Appendix E). Negation perturbations, applied to nearly all records, exhibit a pronounced effect on the BL score, with SP scores showing the highest sensitivity to negation changes. Operator swap perturbations predominantly affect LE scores, which is expected due to LE's reliance on operator structures. Among text-based metrics, BL is the most sensitive to operator perturbations.

Ideally, text-based perturbations should influence translation metric scores, and this is evident in the case of operator and variable perturbations. In contrast, predicate perturbations cause only a minimal drop in scores, as they impact a smaller portion of the dataset, as outlined in Table 1.

RMSE	BL	LE	RO	ME	BS	SP
BLEU	0.90	0.85	0.71	0.78	*0.66*	0.79
LE	0.85	1.05	0.78	0.83	*0.76*	0.84
ROUGE	0.71	0.78	0.69	0.69	**0.60**	0.79
METEOR	0.78	0.83	0.69	0.81	*0.66*	0.76
BERTScore	0.66	0.76	**0.60**	0.66	**0.64**	0.73
Smatch++	0.79	0.84	0.79	0.76	*0.73*	0.83
gpt-4o: 0.86			**o1-preview**: 0.69			

Table 4: Results on the alignment of metric-based and LLM-based ranking of 87 records (each having 1 ground-truth and 3 FOL candidates) against the 3 human annotators consensus. Diagonal values (highlighted) show individual metrics vs. human rankings; off-diagonal values show combined metrics vs. human.

Sample Analysis. Human rankings are used to compare against metric rankings. The inter-annotator agreement between the three annotators, measured using Kendall's tau, shows a correlation of 0.35. We use Root Mean Square Error (RMSE) to evaluate the alignment between human preferences and metric scores. As shown in Table 4, Bertscore demonstrates a stronger alignment with human rankings, even surpassing `o1-preview`. LE score shows the weakest alignment, indicating the importance of semantics for evaluation of FOL statements. The results suggest that, despite the low alignment of structured evaluators such as LE score and Smatch++, using other metrics alongside help with improving their alignment.

6 Conclusion

This study has explored the effectiveness of various metrics in evaluating the correctness of First-Order Logic (FOL) translations of natural language statements. By carefully analyzing the sensitivity of existing metrics through perturbations of ground-truth FOLs, we identified critical gaps in commonly used metrics. Commonly used FOL metrics such as Logical Equivalency and BLEU scores are not insufficient for handling anomalies in FOL generation. To our surprise even LLM-based evaluations via `gpt-4o` model fell short of alignment with human annotation compared with BertScore and combination metrics, suggesting the need for better-suited metrics for evaluating FOL translations, which is essential for advancing the use of LLMs in logical reasoning tasks. Future work can focus on applying these findings to tasks involving sample-based generation methods.

Limitations

We recognize that GPT models used in our experiments are continually evolving, which may lead to variations in results over time. To manage the computational cost of generating multiple samples, we limited the data sample used in the experiments. This could ideally be extended to a larger dataset or used as a reference for achieving high performance in existing methodologies, but not as a standalone solution. The FOLIO dataset, despite being widely used, may contain errors inherent to human judgement.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Luke Benson, Lucy Sun, Ekaterina Zubova, Yujie Qiao, Matthew Burtell, et al. 2022. Folio: Natural language reasoning with first-order logic. *arXiv preprint arXiv:2209.00840*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Theo X Olausson, Alex Gu, Benjamin Lipkin, Cedegao E Zhang, Armando Solar-Lezama, Joshua B Tenenbaum, and Roger Levy. 2023. Linc: A neurosymbolic approach for logical reasoning by combining language models with first-order logic provers. *arXiv preprint arXiv:2310.15164*.

Juri Opitz. 2023. Smatch++: Standardized and extended evaluation of semantic graphs. *arXiv preprint arXiv:2305.06993*.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Yang Wang. 2023. Logic-lm: Empowering large language models with symbolic solvers for faithful logical reasoning. *arXiv preprint arXiv:2305.12295*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Joaquin Quiñonero-Candela, Ido Dagan, Bernardo Magnini, and Florence D'Alché-Buc. 2006. *Machine Learning Challenges: Evaluating Predictive Uncertainty, Visual Object Classification, and Recognizing Textual Entailment, First Pascal Machine Learning Challenges Workshop, MLCW 2005, Southampton, UK, April 11-13, 2005, Revised Selected Papers*, volume 3944. Springer.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Yuan Yang, Siheng Xiong, Ali Payani, Ehsan Shareghi, and Faramarz Fekri. 2024. Harnessing the power of large language models for natural language to first-order logic translation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6942–6959, Bangkok, Thailand. Association for Computational Linguistics.

Xi Ye, Qiaochu Chen, Isil Dillig, and Greg Durrett. 2024. Satlm: Satisfiability-aided language models using declarative prompting. *Advances in Neural Information Processing Systems*, 36.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

A Data Statistics

We aim to ensure diversity in the data used for this study. The FOLIO training set contains 1001 records with ground truth FOLs. The operators used in these FOLs are noted, and we select a unique combination of operators (regardless of their order) for our dataset. By expanding the data, we observe additional operator combinations for a given sentence. For each set of operators, we generate four sentence variations. The details on the data size are provided in are in Table 5, and the distribution of operators can be referred to in Figure 1.

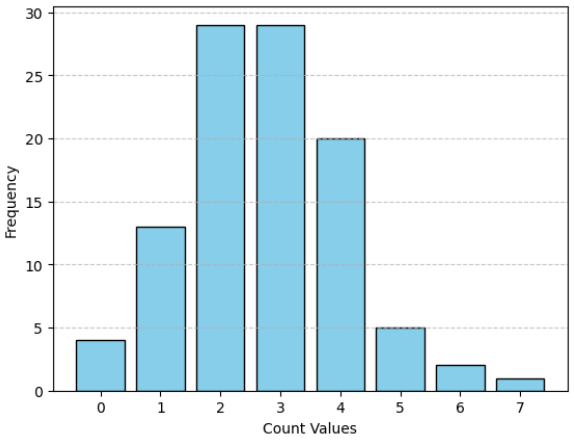

Figure 1: Plot showing the distribution of the operators in each records. The majority of records contain 2-3 operators. Records with 0 operators reflect the presence of a single predicate, indicating no logical connections, while records with 7 operators represent complex statements.

B Sample Generation

We generate samples using the prompt "*Given a natural language sentence, your task is to convert the sentence into first-order logic statements using the following operators:* $\wedge, \vee, \neg, \rightarrow, \leftrightarrow, \forall, \exists, =, \oplus$. *The output is a single first-order statement representing the sentence with no additional tasks. Generate 3 different samples of output.*", where gpt4 provides 3 samples in the form presented in Figure 2.

Size	Operators	Unique	Duplicates
102	$[\wedge \vee \neg \rightarrow \iff \forall \exists = \oplus]$	52	4

Table 5: Dataset statistics used in the sample data for FOLIO dataset

Given a natural language sentence, your task is to convert the sentence into first-order logic statements using the following operators:

$$\wedge, \vee, \neg, \rightarrow, \leftrightarrow, \forall, \exists, =, \oplus.$$

The output is a single first-order statement representing the sentence with no additional tasks. Generate 3 different samples of output.

Text: All eels are fish.
Output: 1. $\forall x(\text{Eel}(x) \rightarrow \text{Fish}(x))$ 2. $\forall x(\text{E}(x) \rightarrow \text{F}(x))$ 3. $\forall x(\text{IsEel}(x) \rightarrow \text{IsFish}(x))$

Figure 2: Example of sample generation using `gpt-4o`. The highlighted text is the output from the LLM.

C Annotator Instruction

The task is to rank the first-order logic (FOL) translations for a given 'gold label' a rank of [1,2,3], where 1 represents the best match and 3 represents a comparatively bad match to the gold FOL. You are given 3 variations of FOL for each sentence. Please feel free to rank based on your preference.

Few good-to-know instructions:

- $F_1 \wedge F_2$: Logical AND, True only if both F_1 and F_2 are true
- $F_1 \vee F_2$: Logical OR, False if both F_1 and F_2 are false
- $\neg$: Negation
- $\rightarrow$: Implies
- $\Leftrightarrow$ Double Implies
- $\forall$: For All quantifier
- $\exists$: There Exists quantifier
- $=$: Equals
- $F_1 \oplus F_2$: XOR, True only if F_1 or F_2 are true

If two FOLs are the 'same', randomly number them. *Ex:* F_1: $A \wedge B$ Rank 3, F_2: $A \wedge B$ Rank 2, F_3: $A \rightarrow B$ Rank 1.

You can lower the rank for structure (syntax) or grammar (semantic) errors. Please do not change the format of the file. Just add the rank next to 'Rank' for each FOL.

Given a ground truth first-order logic statement and three variations of samples, your task is to rank the samples in order of similarity with the label. The output should be a single list with 3 integers, including [1, 2, 3], where 1 represents the closest match and 3 is the least match. Do not include any other explanation and the output form is [rank_sample1, rank_sample2, rank_sample3].

Label: $\forall x(\text{Eel}(x) \rightarrow \text{Fish}(x))$
Sample 1: $\forall x(\text{E}(x) \rightarrow \text{F}(x))$
Sample 2: $\forall x(\text{IsEel}(x) \rightarrow \text{IsFish}(x))$
Sample 3: $\forall x(\text{Eel}(x) \rightarrow \text{Fish}(x))$
Output: [1, 3, 2]

Figure 3: Example of prompt used for ranking the samples using `gpt-4o` and `o1-preview`. The highlighted text is the output from the LLM.

Negation Perturbation	BLEU	LE	ROUGE	METEOR	BERTScore	Smatch++
BLEU	0.69	0.69	0.81	0.77	0.82	**0.52**
LE	0.69	0.73	0.80	0.76	0.82	**0.51**
ROUGE	0.81	0.80	0.93	0.89	0.96	**0.63**
METEOR	0.77	0.76	0.89	0.85	0.90	**0.59**
BERTScore	0.82	0.82	0.96	0.90	0.97	**0.65**
Smatch++	0.52	0.51	0.63	0.59	0.65	**0.37**

Table 6: Negation perturbation scores

AndOr Perturbation	BLEU	LE	ROUGE	METEOR	BERTScore	Smatch++
BLEU	**0.88**	0.77	0.91	0.91	0.93	0.88
LE	0.77	**0.72**	0.81	0.81	0.83	0.78
ROUGE	0.91	**0.81**	0.95	0.95	0.97	0.92
METEOR	0.91	**0.81**	0.95	0.96	0.97	0.92
BERTScore	0.93	**0.83**	0.97	0.97	1.00	0.94
Smatch++	**0.88**	0.78	0.92	0.92	0.94	0.96

Table 7: AndOr perturbation scores

There are one or more correct rankings. In case of 'all incorrect', pick the rank based on the closest match to the gold FOL.

Example (put your ranking at the end of each statement after "Rank"):

- label: ∀ x (Square(x) → Shape(x))
- FOL1: ∀ x (Square(x) → Shape(x)) Rank: 1
- FOL2: ∀x (¬Shape(x) → ¬Square(x)) Rank: 2
- FOL3: ∀ x (Square(x) → Shape(x)) Rank: 3

D LLM Ranker

We generate ranks using the prompt "*Given a ground truth first-order logic statement and three variations of samples, your task is to rank the samples in order of similarity with the label. The output should be a single list with 3 integers including [1, 2, 3], where 1 represents the closest match and 3 is the least match. Do not include any other explanation and the output form is [rank_sample1, rank_sample2, rank_sample3].*", where gpt4o and o1-preview provide a list of ranking in the form presented in Figure 3.

E Pairwise Perturbations

To study the effect of perturbation on the combinations, we obtain sensitivity scores as shown across Table 3 to Table 11. When compared to a single metric, the combination helps with improving the sensitivity of the metric.

F Package Usage

This paper utilizes automatic evaluation metrics and datasets in compliance with their respective licenses. Specifically, we employ BLEU, BertScore (MIT License), ROUGE (Apache-2.0 License), METEOR (MIT License), Logical Equivalence (Apache-2.0 License), and Smatch++ (GNU General Public License). The dataset FOLIO, used in this research, is open-sourced under the CC-BY-SA-4.0 license.

The packages used in this paper are primarily sourced from the evaluation metrics provided by Hugging Face's Evaluate library. Additionally, the source code for Logical Equivalence and Smatch++ was utilized.

OrXor Perturbation	BLEU	LE	ROUGE	METEOR	BERTScore	Smatch++
BLEU	0.95	**0.91**	0.96	0.96	0.97	0.93
LE	0.91	0.92	0.92	0.92	0.93	**0.89**
ROUGE	0.96	**0.92**	0.98	0.98	0.99	0.95
METEOR	0.96	**0.92**	0.98	0.98	0.98	0.95
BERTScore	0.97	**0.93**	0.99	0.98	1.00	0.95
Smatch++	0.93	**0.89**	0.95	0.95	0.95	0.99

Table 8: OrXor perturbation scores

Operator Perturbation	BLEU	LE	ROUGE	METEOR	BERTScore	Smatch++
BLEU	**0.20**	0.39	0.37	0.31	0.54	0.32
LE	**0.39**	0.62	0.56	0.50	0.73	0.52
ROUGE	**0.37**	0.56	0.53	0.48	0.71	0.49
METEOR	**0.31**	0.50	0.48	0.42	0.65	0.44
BERTScore	**0.54**	0.73	0.71	0.65	0.89	0.66
Smatch++	**0.32**	0.52	0.49	0.44	0.66	0.44

Table 9: Operator perturbation scores

Predicate Perturbation	BLEU	LE	ROUGE	METEOR	BERTScore	Smatch++
BLEU	0.94	0.90	0.95	0.96	**0.97**	0.90
LE	0.90	0.93	0.92	0.92	0.93	0.86
ROUGE	0.95	0.92	0.97	0.97	**0.98**	0.91
METEOR	0.96	0.92	0.97	0.98	0.98	0.91
BERTScore	0.97	0.93	0.98	0.98	**1.00**	0.92
Smatch++	0.90	0.86	0.91	0.91	0.92	0.92

Table 10: Predicate perturbation scores

Variable Perturbation	BLEU	LE	ROUGE	METEOR	BERTScore	Smatch++
BLEU	**0.28**	0.61	0.51	0.47	0.59	0.49
LE	**0.61**	1.00	0.84	0.80	0.92	0.82
ROUGE	**0.51**	0.84	0.74	0.70	0.82	0.72
METEOR	**0.47**	0.80	0.70	0.68	0.79	0.68
BERTScore	**0.59**	0.92	0.82	0.79	0.92	0.81
Smatch++	**0.49**	0.82	0.72	0.68	0.81	0.76

Table 11: Variable perturbation scores