Verification of NLP Model Outputs: Logical Consistency and Fact Verification

Group Name: Checkers

Mohit Khatri
862547409
mkhat023@ucr.edu

Justin Chiu
862286201
jchiu030@ucr.edu

Abstract

Recent advances in large language models (LLMs) have led to impressive performance on various NLP tasks, yet their outputs can still suffer from logical inconsistencies and factual errors. We propose a unified verification framework that employs paraphrase-based consistency checking and external fact verification to assess LLM responses. Our prototype uses a T5-based paraphraser, and SBERT for semantic similarity computation, along with Wikipedia-based fact-checking. Preliminary experiments on 10 questions show that our tool correctly identifies consistency in 6 out of 8 cases and verifies factual accuracy correctly in most instances, despite challenges due to sentence framing and retrieval. These results indicate potential for further refinement to ensure reliable, accurate LLM outputs.

Our prototype is publicly available at: https://github.com/MohitKhatri28/CS206_Prototype

CCS Concepts

• Computing methodologies → Natural language processing; Information retrieval; Machine learning approaches;

• Software and its engineering → Software verification and validation; Software testing and debugging;

Keywords

Natural language processing, logical consistency, fact verification, large language models, NLP robustness, automated verification

1. Introduction

Large language models (LLMs) such as GPT-4 and BERT have transformed natural language processing by achieving state-of-the-art results in tasks like text generation, summarization, and question answering. However, these models still struggle with logical inconsistencies and factual inaccuracies that can undermine their reliability in critical applications. This project proposes a unified verification framework designed to evaluate and improve the robustness of LLM outputs. The framework operates on two complementary fronts: assessing logical consistency through paraphrase generation and semantic similarity, and verifying factual correctness by cross-referencing model responses with external knowledge sources such as Wikipedia. Our prototype implements these ideas using a T5-based paraphraser to generate multiple variants of input questions, a Falcon 7B model to produce answers, and an SBERT model to compute similarity scores. By comparing the responses to paraphrased queries and verifying key claims against Wikipedia content, our system provides automated pass/fail evaluations. This unified approach not only highlights inconsistencies and inaccuracies but also offers a foundation for further refining and training LLMs for more reliable performance.

2. Related Work

Research on NLP verification can be broadly categorized into logical consistency checking and fact verification. Early work focused on logical consistency, where [1] introduced a framework to quantify logical transitivity, commutativity, and negation invariance in LLM outputs. This work revealed that models often struggle to maintain coherent reasoning, particularly in maintaining transitivity as the complexity of input increases. Building on this, [2] developed adversarial attacks to expose logical inconsistencies, emphasizing vulnerabilities in models when faced with semantically altered prompts. Meanwhile, [3] highlighted the need for automated detection mechanisms by analyzing contradictions in AI-generated content, suggesting that traditional NLP models lack robustness in maintaining consistent outputs.

In parallel, the field of fact verification evolved, aiming to ensure factual accuracy in generated content. [4] introduced the TruthfulQA benchmark, designed to evaluate whether models could resist generating plausible-sounding but false information. This benchmark highlighted the prevalence of hallucinations in language models when exposed to misleading queries. Complementing this, [5] provided a comprehensive survey of automated fact-checking methods, outlining frameworks that include claim detection, evidence retrieval, and claim verification. This work emphasized the role of external knowledge sources, such as Wikipedia and fact-checking APIs, in improving verification accuracy, while also discussing the challenges faced by fact-checking systems in handling generative AI outputs, advocating for scalable verification mechanisms adaptable to dynamic content.

An important development in consistency evaluation was the introduction of resources like PARAREL, which consists of cloze-style English paraphrases designed to test models' consistency across paraphrased queries. Research utilizing PARAREL demonstrated that models often fail to provide consistent responses to semantically equivalent inputs, suggesting limitations in their knowledge representation capabilities. Additionally, studies on scientific fact-checking, such as those by [6], have highlighted the complexity of verifying domain-specific content, where models must synthesize information from intricate scientific texts. Techniques employed include sparse retrieval methods like TF-IDF and BM25, alongside dense retrieval using vector representations.

Together, these studies reveal that while significant progress has been made in both logical consistency checking and fact verification, the research remains fragmented. Existing methods either focus on consistency without verifying factual correctness or emphasize fact-checking without considering logical coherence. This fragmentation underscores the need for a unified framework capable of addressing both aspects to enhance the robustness of NLP models.

3. Limitation Analysis

Despite significant progress in NLP verification, existing methods suffer from several limitations:

- Lack of cross-prompt consistency evaluation: Current approaches assess logical stability within fixed datasets, failing to handle dynamic user interactions where paraphrased queries should yield consistent responses.
- Absence of integrated fact verification: Most fact-checking techniques rely on static datasets without effectively incorporating verification into the model evaluation pipeline.
- Fragmented research focus: Logical consistency checking and fact verification are treated as separate problems, with no comprehensive framework integrating both aspects.

These limitations highlight the need for a unified approach that simultaneously ensures logical coherence and factual accuracy in NLP model outputs.

4. Proposed Approach

To address the shortcomings of existing research, we propose an NLP Verification Framework that integrates logical consistency analysis and fact validation. Our approach consists of three key components:

1. Paraphrase-Based Logical Consistency Checking: A paraphrasing model (such as T5) generates reworded versions of test queries, and a similarity metric (e.g., SBERT or cosine similarity) is used to compare AI-generated responses for consistency.
2. Fact Verification via External Knowledge Retrieval: AI-generated responses are validated against external knowledge sources such as Wikipedia or structured databases, using retrieval-augmented verification to assess factual correctness.
3. Evaluation Pipeline: The system flags inconsistencies and misinformation, generating structured outputs to aid in model debugging and improvement.

5. Prototype Implementation

Our prototype is designed as a modular Python-based system that integrates three major components to evaluate the robustness of NLP model outputs: paraphrase generation, response evaluation for logical consistency, and fact verification.

First, for paraphrase generation, we use a T5-based model fine-tuned for paraphrasing (using the model "humarin/chatgpt_paraphraser_on_T5_base"). Given a seed question, the system prefixes it with a specific prompt (e.g., "paraphrase:") and generates multiple paraphrased variants. These paraphrases serve to probe the model under test by rephrasing the original question, allowing us to assess if the model's responses remain consistent regardless of variations in phrasing.

Next, for response evaluation, we deploy a Falcon 7B instruct model as our QA system. This model is queried with both the original seed question and its paraphrased versions. The outputs are then cleaned to remove extraneous tokens and artifacts. To measure logical consistency, we leverage an SBERT model ("all-mpnet-base-v2") to compute semantic similarity between the seed answer and each paraphrased answer. We define a similarity threshold and consider a test case as passing the consistency check if a sufficient number of paraphrased responses exceed this threshold. The system outputs a simple pass/fail result based on whether the majority of paraphrased responses align closely with the seed answer.

The fact verification module extracts key claims from the model's answer. The prototype reconstructs the claim when necessary and queries Wikipedia to fetch relevant content. It then splits the retrieved content into manageable chunks and uses SBERT to find the best matching excerpt. A similarity score is computed between the claim and the selected chunk; if this score meets or exceeds a preset threshold, the claim is considered factually verified. Otherwise, it is flagged as incorrect.

All these steps are implemented using state-of-the-art libraries such as Hugging Face's Transformers for model management, Sentence-Transformers for semantic similarity, and PyTorch for efficient computation (leveraging GPU when available). The final output from the evaluation pipeline is a set of pass/fail results for both logical consistency and fact verification, enabling clear identification of any shortcomings in the model's responses. This structured feedback—while simple—is a critical step towards building systems that can automatically diagnose and improve the reliability of LLM-generated content.

6. Experimental Results

We conducted preliminary experiments using our verification framework on a set of 10 diverse questions. For each question, the system generated five paraphrased variants using a T5-based model to assess logical consistency, while the Falcon 7B model produced answers for both the original and paraphrased questions. Semantic similarity between the seed answer and its paraphrased counterparts was then computed using an SBERT model with a threshold set at 0.6.

In the logical consistency evaluation, the model generated consistent responses for 8 out of the 10 questions. Our tool correctly classified 6 of these 8 cases as consistent and accurately flagged the remaining 2 instances as inconsistent. These results suggest that while the method is effective at identifying clear inconsistencies, subtle variations in phrasing can sometimes lead to lower similarity scores and potential misclassifications, even when the responses are semantically equivalent.

For fact verification, our module extracts key claims from the model's answers and retrieves related content from Wikipedia for validation, using a similarity threshold of 0.65. Although the model generated factually correct responses in 9 out of 10 cases, the tool only verified 6 of these as true. Notably, the tool occasionally struggled to extract the most relevant excerpt from the Wikipedia summary. Consequently, even when the model's answer was factually correct, an irrelevant or incomplete excerpt led to a low similarity score, causing the tool to erroneously flag the answer as incorrect.

Overall, these experiments underscore both the potential and the challenges of our framework. The logical consistency component generally performs well, though it is sensitive to nuanced phrasing differences. Meanwhile, the fact verification module demonstrates promise but requires further refinement in its retrieval process to ensure that the most pertinent Wikipedia content is identified. Addressing these issues could significantly enhance the framework's reliability for verifying LLM outputs.

7. Conclusion

In conclusion, our unified verification framework demonstrates the potential to automatically identify both logical inconsistencies and factual inaccuracies in LLM outputs. Through our prototype implementation, we were able to assess responses across multiple paraphrased queries and verify key factual claims against Wikipedia. Experimental results on 10 questions indicate that the system reliably classifies consistency in the majority of cases and correctly flags erroneous outputs. However, challenges remain, particularly in the fact verification process where the retrieval of relevant Wikipedia excerpts sometimes leads to misclassifications despite the model's correct responses. These findings highlight the need for further refinements, such as improved text retrieval techniques and dynamic threshold adjustments. Ultimately, this work offers a promising step toward enhancing the robustness and reliability of LLM-generated content, with significant implications for deploying these models in real-world applications where accuracy and consistency are paramount.

References

[1] Yinhong Liu, Zhijiang Guo, Tianya Liang, Ehsan Shareghi, Ivan Vulić, and Nigel Collier. 2024. Aligning with Logic: Measuring, Evaluating and Improving Logical Consistency in Large Language Models. Retrieved from https://arxiv.org/abs/2410.02205

[2] Mutsumi Nakamura, Santosh Mashetty, Mihir Parmar, Neeraj Varshney, and Chitta Baral. 2023. LogicAttack: Adversarial Attacks for Evaluating Logical Consistency of Natural Language Inference. In Findings of the Association for Computational Linguistics: EMNLP 2023, Association for Computational Linguistics, Singapore, 13322–13334. DOI:https://doi.org/10.18653/v1/2023.findings-emnlp.889

[3] Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. 2021. Measuring and Improving Consistency in Pretrained Language Models. Retrieved from https://arxiv.org/abs/2102.01017

[4] Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring How Models Mimic Human Falsehoods. Retrieved from https://arxiv.org/abs/2109.07958

[5] Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. A Survey on Automated Fact-Checking. Retrieved from https://arxiv.org/abs/2108.11896

[6] Juraj Vladika and Florian Matthes. 2023. Scientific Fact-Checking: A Survey of Resources and Approaches. Retrieved from https://arxiv.org/abs/2305.16859

[7] Prototype: https://github.com/RedFiringSun/Verification-of-NLP/tree/main