SelfCheckGPT: Implementation and Experimental Analysis

Course Project Report

Mohamed Dhifallah

15 August 2025

Abstract

This report presents the re-implementation and experimental evaluation of the SelfCheckGPT method for hallucination detection in large language models. The work follows the methodology proposed by Manakul et al. (2023) and focuses on building a modular, resource-conscious framework that implements multiple scoring strategies, including an n-gram language model, BERTScore, natural language inference (NLI), multi-question answering (MQAG), and a prompt-based LLM judge. Experiments are conducted on the WikiBio hallucination dataset, replicating and extending aspects of the original study while remaining within computational constraints. The results show that a combined multi-metric approach achieves superior performance compared to any single metric, confirming the core findings of the original paper.

Contents

1 Introduction

Hallucination detection has become a critical challenge in the deployment of large language models (LLMs), especially when accessed as black-box services. SelfCheckGPT offers a zero-resource solution that relies solely on repeated sampling from the target LLM, measuring the consistency of generated outputs as a proxy for factual accuracy. This report documents the design, implementation, and evaluation of a modular framework inspired by SelfCheckGPT, focusing on reproducibility and extensibility under constrained computational budgets.

2 Summary of the Original Paper

The original SelfCheckGPT paper by Manakul et al. (2023) introduces a black-box hallucination detection method that samples multiple completions for each input prompt and scores the factuality of each sentence based on its support across these samples. Three main contributions are highlighted:

- **Methodology:** A fact-consistency score requiring no external knowledge base, computed from intra-model agreement.
- **Experimental validation:** Evaluation on GPT-3 outputs from the WikiBio dataset, with sentence- and passage-level metrics.
- **Baseline comparison:** Demonstration that SelfCheckGPT outperforms uncertainty-based baselines.

The paper reports that factual sentences tend to appear consistently across samples, while hallucinations exhibit higher variability or contradictions.

3 Implementation

The framework developed for this project is implemented in Python and organized around a central experiment runner. It incorporates the following metrics:

1. **SelfCheckNgram:** Unigram LM trained on sampled passages, scoring sentences by likelihood.
2. **SelfCheckBERTScore:** Semantic similarity between candidate sentences and sampled passages using RoBERTa-Large.
3. **SelfCheckNLI:** DeBERTa-large MNLI model to classify entailment vs. contradiction.
4. **SelfCheckMQAG:** Multi-question generation and answering pipeline to assess factual consistency.
5. **SelfCheckPrompt:** Prompt-based LLM judge returning a binary support decision.

Resampling is handled via an OpenAIChatLLM interface, with options for temperature, top-k, top-p, and deterministic sampling. Evaluation metrics include Average Precision (AP), F1-score, precision, recall, and Brier score.

4 Experimental Setup

Experiments were conducted on the WikiBio hallucination dataset. Following the paper, 20 samples per prompt were generated using the `-paper-config` settings (temperature 0.7, top-k 50, top-p 0.95). The setup included:

- **Single-metric evaluation:** Each metric scored independently.
- **Threshold tuning:** Thresholds maximizing F1 on training split applied to test split.
- **Combined model:** Logistic regression ensemble of all metrics.
- **Small-scale tests:** Limited to 20 examples for rapid iteration.

5 Results and Discussion

Table 1 presents the results for the 20-example run. The combined model achieved the highest AP and balanced F1, outperforming individual metrics.

Table 1: Performance on a 20-example run (thresholds tuned on train split).

Metric	AP	Brier	F1	Precision	Recall
N-gram	0.814	0.265	0.855	0.756	0.984
NLI	0.868	0.467	0.852	0.752	0.984
BERTScore	0.733	0.724	0.850	0.744	0.992
LLM-Prompt	0.754	0.263	0.851	0.755	0.976
Combined	**0.871**	**0.189**	**0.859**	**0.758**	**0.992**

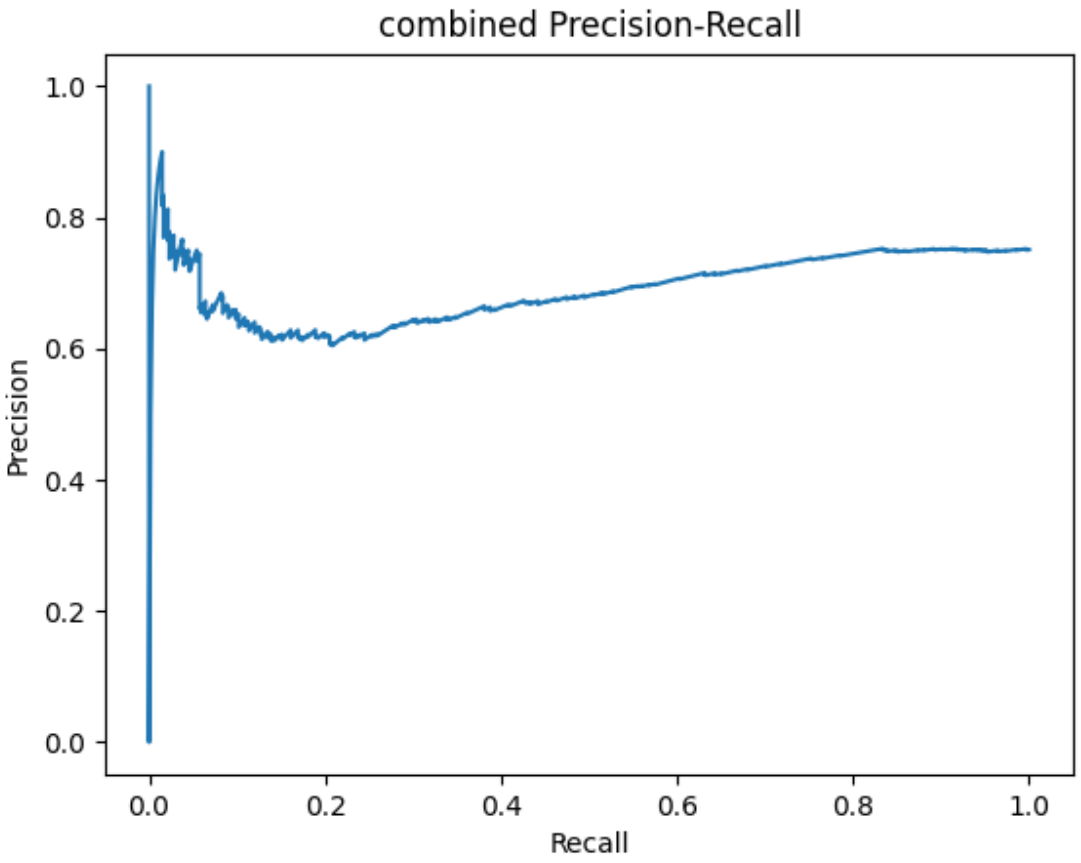

Figure 1: Precision–Recall curves for combined and individual metrics.

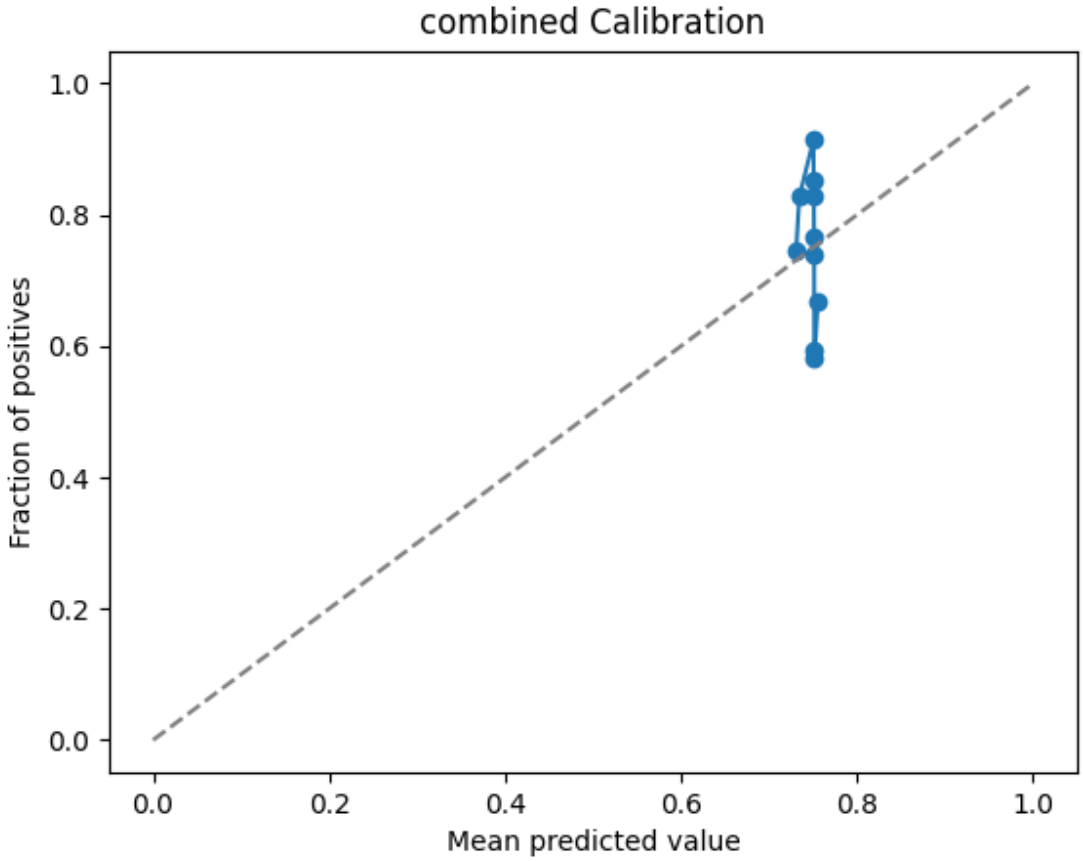

Figure 2: Calibration plots for combined and individual metrics.

6 Contributions

This project's key contributions include:

- Modular reimplementation of SelfCheckGPT scoring metrics.
- Offline HuggingFace backend for prompt-based scoring.
- Threshold tuning for optimal F1 performance.
- Demonstration of qualitative trends from the original study under resource constraints.

7 Limitations

- Consistency-based scoring may fail on confidently wrong statements.
- Results limited by small-scale experiments and dataset domain.
- Some metrics (e.g., MQAG) may vary due to sampling heuristics.

8 Conclusion

The reimplementation confirms the original insight: combining multiple lightweight signals improves hallucination detection in black-box LLMs. While limited in scope, the experiments support the method's general applicability.

9 Personal Reflections

Engaging with SelfCheckGPT reinforced my belief in the value of simple, interpretable heuristics in AI safety. The method's elegance lies in leveraging the model's own variability as a diagnostic signal. While the results were encouraging, they also highlighted areas where future work could stress-test the method's robustness, such as in diverse domains or with adversarial inputs. I appreciated the clarity of the original writing and found it straightforward to translate into a working system, even if certain calibration details required experimentation. Overall, this project deepened my understanding of practical hallucination detection strategies for large language models.