FactSelfCheck: Fact-Level Black-Box Hallucination Detection for LLMs

Albert Sawczyn[1] **Jakub Binkowski**[1] **Denis Janiak**[1]

Bogdan Gabrys[2] **Tomasz Kajdanowicz**[1]

[1]Wrocław University of Science and Technology
[2]University of Technology Sydney
albert.sawczyn@pwr.edu.pl

Abstract

Large Language Models (LLMs) frequently generate hallucinated content, posing significant challenges for applications where factuality is crucial. While existing hallucination detection methods typically operate at the sentence level or passage level, we propose FactSelfCheck, a novel zero-resource black-box sampling-based method that enables fine-grained fact-level detection. Our approach represents long-form text as interpretable knowledge graphs consisting of facts in the form of triples, providing clearer insights into content factuality than traditional approaches. Through analyzing factual consistency across multiple LLM responses, we compute fine-grained hallucination scores without requiring external resources or training data. Our evaluation demonstrates that FactSelfCheck performs competitively with leading sentence-level sampling-based methods while providing more detailed and interpretable insights. Most notably, our fact-level approach significantly improves hallucination correction, achieving a 35.5% increase in factual content compared to the baseline, while sentence-level SelfCheckGPT yields only a 10.6% improvement. The granular nature of our detection enables more precise identification and correction of hallucinated content. Additionally, we contribute FavaMultiSamples, a novel dataset that addresses a gap in the field by providing the research community with a second dataset for evaluating sampling-based methods.

1 Introduction

Large Language Models (LLMs) have gained significant attention from academia and industry recently. However, a major limitation of LLMs is their tendency to generate hallucinated information (Farquhar et al., 2024; Huang et al., 2025), posing significant challenges for applications where factual correctness is crucial, such as healthcare (Sallam, 2023). Although numerous methods have been proposed to reduce hallucinations (Zhang et al., 2023), it is not possible to eliminate them, and LLMs will constantly hallucinate (Lee, 2023; Xu et al., 2024). Therefore, there remains a critical need for reliable hallucination detection in LLM responses, particularly for long-form text generation tasks where complexity and information density increase the risk of factual errors. Effective detection enables system interventions by either preventing the transmission of hallucinated content to users or facilitating its correction (Zhang et al., 2023).

Previous approaches to hallucination detection have primarily focused on classifying hallucinations at either the passage or sentence level (Huang et al., 2025). While valuable, these approaches are limited in their granularity and interpretability, as they do not provide detailed information about specific hallucinated facts or clear insights into what exactly is wrong. This limitation becomes particularly evident in long-form text generation. To address this limitation, we propose a novel method for hallucination detection that operates at the fact level, offering finer-grained and more interpretable analysis. In our approach, we define a fact as a triple consisting of a head, relation, and tail – a standard representation in knowledge graphs (e.g., (*Robert Smith*, *member of*, *The Cure*)) (Hamilton et al., 2017). Our method provides more precise, actionable, and interpretable information by computing hallucination scores for individual facts than traditional passage-level or sentence-level classification approaches. This structured representation, through knowledge graphs, provides enables straightforward interpretation and verification (Pan et al., 2024).

Our granular approach is motivated by two key observations. First, a single sentence can contain multiple facts, with the number of facts varying significantly across sentences, contexts, and domains. This variability makes it challenging to identify hallucinated aspects of generated output precisely

Findings of the Association for Computational Linguistics: EACL 2026, pages 5603–5621
March 24-29, 2026 ©2026 Association for Computational Linguistics

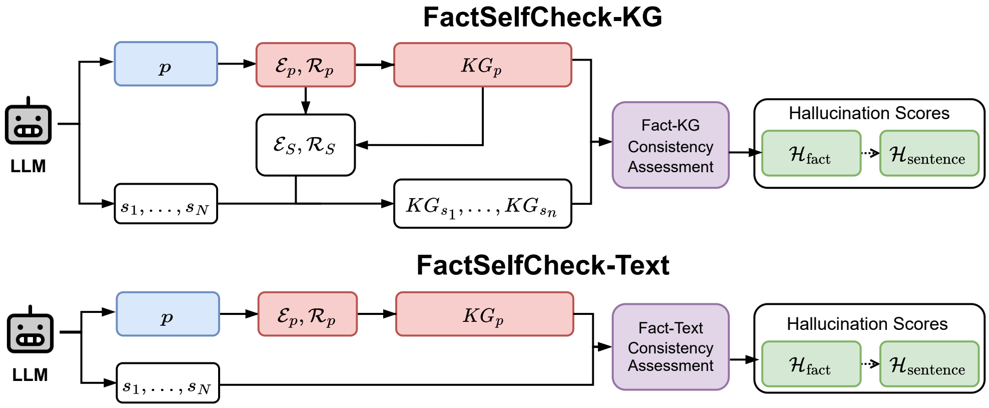
FactSelfCheck-KG
LLM
p
$\mathcal{E}_p, \mathcal{R}_p$
KG_p
$\mathcal{E}_S, \mathcal{R}_S$
$s_1, \ldots, s_N$
$KG_{s_1}, \ldots, KG_{s_n}$
Fact-KG Consistency Assessment
Hallucination Scores
$\mathcal{H}_{\text{fact}}$
$\mathcal{H}_{\text{sentence}}$
FactSelfCheck-Text
LLM
p
$\mathcal{E}_p, \mathcal{R}_p$
KG_p
$s_1, \ldots, s_N$
Fact-Text Consistency Assessment
Hallucination Scores
$\mathcal{H}_{\text{fact}}$
$\mathcal{H}_{\text{sentence}}$

Figure 1: The pipeline of FactSelfCheck in two variants. For response p, entities $\mathcal{E}_p$ and relations $\mathcal{R}_p$ are extracted, followed by the construction of knowledge graphs KG_p, for which hallucination scores $\mathcal{H}_{\text{fact}}$ are calculated. Samples' entities $\mathcal{E}_S$ and relations $\mathcal{R}_S$ are created by merging $\mathcal{E}_p$ and $\mathcal{R}_p$ with entities and relations from KG_p. For each sample s, the knowledge graph KG_s is extracted. FactSelfCheck-KG assesses the consistency between a fact and all KG_s. FactSelfCheck-Text assesses the consistency between a fact and all s directly. To obtain sentence-level score $\mathcal{H}_{\text{sentence}}$, fact-level scores are aggregated, as indicated by dashed arrows.

when using sentence-level detection. Second, false information can be dispersed throughout a text, as a single fact may appear across multiple sentences. That can mislead the sentence-level detection, as the factuality of a sentence is dependent on the previous sentences (Zhang et al., 2024a). Fine-grained fact-level detection provides a more precise understanding of text factuality than a sentence analysis. It enables better assessment of content reliability and, as we show later, more effective factuality correction.

We propose FactSelfCheck, a black-box method for fact-level hallucination detection, meaning it does not require access to the model's internal parameters. This design choice makes our approach universally applicable across any LLM, including closed models, like GPT (OpenAI et al., 2024). Following a sampling-based detection paradigm, introduced by Manakul et al. (2023), our method utilizes multiple response generations and analyzes the factual consistency of extracted facts across these samples. This paradigm is based on the phenomenon that factual information remains largely consistent across different generations, while hallucinated content tends to vary or contradict itself between samples (Manakul et al., 2023; Wang et al., 2023). This way, we can effectively identify hallucinated facts without relying on external resources (zero-resource) or access to the model's internal parameters (black-box). Our zero-resource, non-parametric approach requires no external knowledge bases or training data, making it broadly applicable across domains. The FactSelfCheck pipeline consists of three main steps: knowledge graph extraction, which extracts sets of facts from the initial response and samples; fact-level hallucination scoring; and calculating sentence-level scores by aggregating fact-level scores.

We evaluated our method using the WikiBio GPT-3 Hallucination Dataset (Manakul et al., 2023) and the FavaMultiSamples dataset. WikiBio was the only existing dataset for evaluating sampling-based hallucination detection methods. To address this significant gap, we developed FavaMultiSamples, providing an additional evaluation benchmark. We performed both sentence-level and fact-level evaluation, with sentence-level scores aggregated from fact-level scores. Our approach achieves performance comparable to leading sampling-based methods at the sentence level while providing more detailed information about hallucinations. We also demonstrate effective fact-level hallucination detection and show that our fact-level approach significantly improves hallucination correction. Compared to a baseline, providing incorrect facts to the correction method leads to a 35.5% increase in factual content, while passing incorrect sentences leads to an 10.6% increase.

Our key contributions are as follows:

1. The novel zero-resource black-box sampling-based method for fact-level hallucination detection – FactSelfCheck, designed for long-form text generation. It enables fine-grained hallucination detection in LLM responses without requiring training data or external

resources, as it is both non-parametric and zero-resource. We propose two effective approaches for measuring factual consistency across multiple samples: FactSelfCheck-KG using knowledge graph comparisons and FactSelfCheck-Text using direct text comparison.

2. The FavaMultiSamples dataset, a novel dataset for evaluating sampling-based methods.

3. Comprehensive evaluation of our method, which shows competitive performance with leading sampling-based methods while providing more detailed insights.

4. Demonstration that fact-level detection significantly improves hallucination correction compared to sentence-level approaches.

Our code is available on GitHub [1]. The FavaMultiSamples dataset is available on Hugging Face [2]. We publish all the code and data, allowing for the reproduction of the results.

2 Related work

Xu et al. (2024) have proven that hallucinations are inevitable in LLMs. As LLMs are powerful tools, many recent studies have been conducted regarding hallucination mitigation and detection (Zhang et al., 2023; Huang et al., 2025). The detection methods can be divided into two groups: white-box and black-box.

White-box methods analyze LLMs' internal states (Farquhar et al., 2024; Azaria and Mitchell, 2023). While these methods are universal across all LLMs, they often require multiple generations, similar to sampling-based methods. Notable approaches include: SAPLMA (Azaria and Mitchell, 2023), which predicts from hidden states whether generated text is correct or incorrect; INSIDE (Chen et al., 2024), which evaluates hidden state consistency across generations; SEPs (Kossen et al., 2024) that predict entropy directly from model hidden states; Lookback Lens (Chuang et al., 2024) and AttentionScore (Sriramanan et al., 2024) that uses attention maps to detect hallucinations.

Black-box approaches operate without access to the model's internal states and aim to detect hallucinations based solely on the text generated by LLMs. Some of these methods use external resources to collect evidence (Min et al., 2023; Chern et al., 2023). Others leverage LLMs to detect hallucinations like CoVe (Dhuliawala et al., 2024), which utilizes the chain-of-thought paradigm for detection. Another category is sampling-based methods, such as SelfCheckGPT (Manakul et al., 2023), which evaluate factuality by generating multiple responses (stochastic samples) and assessing consistency between the original response and these samples. The paradigm of utilizing LLM to check its own responses was widely studied and adopted in many works (Kadavath et al., 2022; Lin et al., 2024; Ferraz et al., 2024; Zhang et al., 2024b; Miao et al., 2023). Many of these approaches employ a multi-step decomposition strategy to break down the complex task of hallucination detection into more manageable subtasks, a methodology we also adopt in our approach.

The most popular approach is to classify hallucinations at sentence-level or passage-level (Huang et al., 2025). Few methods have been specifically designed to detect hallucinations at the fact level, i.e. where facts are defined as triples. GraphEval (Sansford et al., 2024) generates a KG from LLM output and compares it with the context provided in the LLM input. FactAlign (Rashad et al., 2024) builds KGs from LLM output and source text, then compares them after performing entity alignment, a technique that pairs the same entity in different KGs. Knowledge-centric detection (Hu et al., 2024) similarly extracts knowledge triplets for fine-grained detection. All these methods require external context as reference, while our method is designed to work without any external knowledge sources by leveraging sampling-based consistency analysis. When evaluating on well-established sources like Wikipedia, methods with access to source materials may achieve superior performance (Manakul et al., 2023). However, our zero-resource approach offers broader applicability, as it can be applied to any task without requiring external knowledge sources.

Most similar to our approach is GCA (Fang et al., 2024), which constructs KGs from the response and samples and then compares them by aggregating multiple scores. However, GCA has significant methodological concerns – they tuned 6 hyperparameters directly on the evaluation set (the only

[1] github.com/graphml-lab-pwr/FactSelfCheck license: CC BY-SA 4.0

[2] huggingface.co/datasets/graphml-lab-pwr/FavaMultiSamples license: CC BY-SA 4.0

available split in the WikiBio GPT-3 hallucination dataset (Manakul et al., 2023)), making it methodologically problematic. Due to these concerns, we cannot provide a fair quantitative comparison with GCA. In contrast, FactSelfCheck is truly zero-shot and parameter-free, requiring no parameter tuning. We achieved that by designing a constrained KG extraction that works consistently across multiple generation samples, rather than using freeform extraction like GCA.

3 Method

We propose FactSelfCheck, a black-box sampling-based method for fact-level hallucination detection, as illustrated in Figure 1. Our method is specifically designed for long-form text generation scenarios, where passages typically contain several sentences with complex factual information.

3.1 Notation

Let p denote the initial response passage generated by the LLM to a user query, which we aim to evaluate for hallucinations. Let $S = \{s_1, \ldots, s_N\}$ represent a set of N stochastic LLM response samples. The text passage p consists of a set of sentences U. For each sentence $u \in U$ and each sample $s \in S$, we extract knowledge graphs KG_u and KG_s, respectively. Each knowledge graph comprises a set of facts, where a fact f is defined as a triple (h, r, t) consisting of a head h, relation r, and tail t, e.g. (*Robert Smith*, *member of*, *The Cure*). We define $KG_p = \bigcup_{u \in U} KG_u$ as the knowledge graph consisting of all facts from the passage p.

Our objective is to compute a fact-level hallucination score $\mathcal{H}_{\text{fact}}$ for each fact f in KG_p. Subsequently, to facilitate comparisons with other methods, we aggregate these scores to obtain a sentence-level hallucination score $\mathcal{H}_{\text{sentence}}$ for each sentence u.

3.2 FactSelfCheck pipeline

As shown in Figure 1, the pipeline of FactSelfCheck consists of three main steps: (1) **Knowledge Graph Extraction** that extracts sets of entities, relations, and finally, knowledge graph from the initial response p and samples S; (2) **Fact-level Hallucination Scoring**, that score facts by measuring factual consistency between facts in KG_p and, depending on the variant, KG_s in FactSelfCheck-KG or directly s in FactSelfCheck-Text; (3) **Sentence-Level Score** calculation by aggregation of the fact-level scores.

3.3 Knowledge Graph Extraction

We adopt an approach that decomposes the knowledge graph extraction task into simpler subtasks, similarly to Edge et al. (2024). This process involves three primary steps: extracting entities, identifying relations, and formulating facts, which are implemented as a sequence of LLM prompts.

For each instance, we extract a list of entities $\mathcal{E}_p$ by passing p to the LLM.

$$\mathcal{E}_p = LLM_{\text{entities}}(p) \tag{1}$$

Next, we provide p and $\mathcal{E}_p$ to the LLM to extract relation types between entities, resulting in $\mathcal{R}_p$.

$$\mathcal{R}_p = LLM_{\text{relations}}(p, \mathcal{E}_p) \tag{2}$$

We then input $\mathcal{E}_p$, $\mathcal{R}_p$, and each sentence $u \in U$ into the LLM to extract the knowledge graph KG_u. We also provide an initial response p to add contextual information. The output is a set of facts:

$$KG_u = LLM_{\text{sentence\_facts}}(u, p, \mathcal{E}_p, \mathcal{R}_p) \tag{3}$$

After extracting KG_u for each sentence $u \in U$, we compile the sets of entities $\mathcal{E}_S$ and relations $\mathcal{R}_S$ required for extracting knowledge graph KG_s from each sample s [3]:

$$\mathcal{E}_S = \mathcal{E}_p \cup \bigcup_{u \in U} \{h, t \mid (h, r, t) \in KG_u\} \tag{4}$$

$$\mathcal{R}_S = \mathcal{R}_p \cup \bigcup_{u \in U} \{r \mid (h, r, t) \in KG_u\} \tag{5}$$

$$KG_s = LLM_{\text{sample\_facts}}(s, \mathcal{E}_S, \mathcal{R}_S) \tag{6}$$

Extracting KG_s by utilizing $\mathcal{E}_S$ and $\mathcal{R}_S$ is more convenient and robust than extraction without them, as it eliminates the need for entity alignment and ensures that the KG is built using the same schema as KG_p.

[3] Although we could theoretically use $\mathcal{E}_p$ and $\mathcal{R}_p$ for KG_s extraction, in practice, LLMs are not sufficiently accurate to extract all entities and relations from the response p when calculating $\mathcal{E}_p$ and $\mathcal{R}_p$. This results in KG_u containing entities and relations not present in $\mathcal{E}_p$ and $\mathcal{R}_p$, even if the prompt restricts them. Empirical tests showed that extending $\mathcal{E}_S$ and $\mathcal{R}_S$ by adding entities and relations from all KG_u improved the results.

3.4 Fact-Level Hallucination Scores

We define two variants for measuring fact-level hallucination scores. The first variant, FactSelfCheck-KG, assesses the consistency between a fact and the knowledge graphs extracted from samples. The second variant, FactSelfCheck-Text, evaluates the consistency between a fact and the samples directly.

3.4.1 FactSelfCheck-KG

In the FactSelfCheck-KG variant, we introduce two metrics to assess the reliability of each fact.

Frequency-Based Hallucination Score The frequency-based fact-level hallucination score is based on the intuition that the probability of a fact being hallucinated is inversely proportional to the fraction of samples containing the same fact.

$$\mathcal{H}_{\text{fact}}(f) = 1 - \frac{1}{|S|}\sum_{s \in S} \mathbb{I}\{f \in KG_s\} \tag{7}$$

where $\mathcal{H}_{\text{fact}}(f)$ is the hallucination score for fact f, and $\mathbb{I}\{f \in KG_{s_n}\}$ is an indicator function that equals 1 if fact f appears in KG_{s_n} and 0 otherwise. The higher the $\mathcal{H}_{\text{fact}}$ value, the higher the plausibility of hallucination.

LLM-Based Hallucination Score To allow semantic matching and reasoning over knowledge graphs, rather than only exact fact matching, we introduce the LLM-based fact-level hallucination score. We instruct the LLM to determine whether each fact is supported by the knowledge graphs extracted from the samples. The LLM is expected to respond with 'yes' or 'no'. We then average the valid responses from the LLM to get the final score as in Equation (8). Any invalid responses are not included in the averaging.

$$\mathcal{H}_{\text{fact}}(f) = \frac{1}{|V_f|}\sum_{s \in V_f} \Psi(f, KG_s) \tag{8}$$

where V_f represents the set of samples with valid LLM responses for the fact f, and the function Ψ is defined as follows:

$$\Psi(\cdot) = \begin{cases} 0 & \text{if the LLM returns 'yes'} \\ 1 & \text{if the LLM returns 'no'} \end{cases} \tag{9}$$

3.4.2 FactSelfCheck-Text

In the FactSelfCheck-Text variant, we check if a fact is supported by each textual sample directly without using the knowledge graphs. We prompt the LLM to evaluate whether a fact f is supported by the textual sample s. As in the previous variant, we average the valid LLM responses using the Ψ function:

$$\mathcal{H}_{\text{fact}}(f) = \frac{1}{|V_f|}\sum_{s \in V_f} \Psi(f, s) \tag{10}$$

3.5 Sentence-Level Hallucination Score

While detecting hallucinations at the fact level offers fine-grained insights, there are scenarios where sentence-level detection is necessary, such as for comparison with existing baselines. To achieve this, we aggregate fact-level scores to compute sentence-level scores $\mathcal{H}_{\text{sentence}}(u)$. This aggregation bridges the gap between atomic fact-level judgments and coarser sentence-level evaluations.

$$\mathcal{H}_{\text{sentence}}(u) = \text{Agg}_{f \in KG_u} \mathcal{H}_{\text{fact}}(f) \tag{11}$$

where u represents a single sentence, and U is the set of sentences of the response p. The aggregation function Agg defines how the factuality of individual facts determines the factuality of the sentence. We employ two distinct aggregation strategies: $mean$ and max. The $mean$ function computes the average hallucination score of all facts within a sentence, providing a smoothed measure of the overall factual density. This is useful for assessing general content quality where partial correctness matters. In contrast, the max function identifies the most severe hallucination within the sentence, based on the intuition that even a single hallucinated fact is sufficient to identify the sentence as hallucinated.

4 Experimental Setup

In this section, we describe the experimental setup, including the used data, implementation details, and aspects we investigated. Additionally, we conducted a fact-level evaluation to provide a more comprehensive analysis. The detailed description of it is provided in Appendix C.2.

4.1 Evaluation Data

Since the method is designed for detecting hallucinations in long generated passages, finding appropriate datasets was challenging. We evaluated our method on two datasets. The first one is the WikiBio GPT-3 Hallucination Dataset

(Manakul et al., 2023) [4] (later referred to as WikiBio). To the best of our knowledge, this was the only dataset specifically designed for evaluating sampling-based hallucination detection methods, representing a significant research gap. To address this, we developed FavaMultiSamples, a novel dataset specifically designed for evaluating methods that analyze multiple samples, providing researchers with an additional benchmark for robust evaluation. We built it upon the FAVA dataset (Mishra et al., 2024) and the detailed description of creation is provided in Appendix A. WikiBio covers generated biographical passages, while FavaMultiSamples includes diverse knowledge-intensive queries across various domains. Both datasets were annotated by humans. The dataset statistics are provided in Appendix B, and statistics related to KG extraction are detailed in Appendix C.4. Both datasets contain only test data, making it methodologically incorrect to tune parameters, including classification thresholds, on these datasets.

Sentence-level While the datasets focus on sentence-level evaluation, our approach provides more fine-grained insights through fact-level analysis. To ensure a meaningful comparison with SelfCheckGPT, we evaluated these levels using aggregation approaches (see Section 3.5). For WikiBio, we followed the protocol established by Manakul et al. (2023), merging the labels *major-inaccurate* and *minor-inaccurate* into a single *hallucination* class.

Fact-level To enable fact-level evaluation, we annotated facts from the extracted response knowledge graph (KG_p) using the LLM-as-judge approach (Zheng et al., 2023). For each fact, we provided the external biography from Wikipedia as a reference. The LLM-as-judge annotated whether each fact is supported by the reference. We utilized GPT-4o for this task, to ensure high quality of the annotations. As a result, we obtained 5488 binary annotated facts.

The reliability of this approach is high, and commonly used in the literature (Thakur et al., 2025). The LLM-as-judge approach has been specifically tested for annotating hallucinations in LLM output given a knowledge source by Janiak et al. (2025) and has demonstrated high alignment with human annotations, outperforming other automated evaluation methods such as ROUGE (Lin, 2004).

4.2 Baseline Models

We compared our method against several key baselines. The RandomSentence baseline predicts a random class for each sentence with equal probability. The RandomFact baseline predicts random scores for each fact and aggregates them to obtain sentence-level scores. The probability-based baselines, proposed by Manakul et al. (2023), use $-\log p$ and $\mathcal{H}$ to measure likelihood and entropy of each token respectively and aggregate the scores using Mean or Max functions to obtain sentence-level scores.

We include the two best-performing variants of SelfCheckGPT: Prompt and NLI. We implemented the Prompt variant using the same LLM employed in our method, namely Llama-3.1-70B-Instruct, whereas the original method utilized an unspecified release of GPT-3.5-turbo. For the NLI variant, we used the same model as was used in the original paper[5]. Finally, the AttentionScore leverages attention maps (Sriramanan et al., 2024), which is, to the best of our knowledge, the only unsupervised internal state-based method. Its unsupervised nature was crucial since the available datasets do not provide training data. We adapted the AttentionScore for sentence-level detection by implementing two variants: (1) Absolute, which analyzes the complete attention map from the input start to the end of the given sentence, and (2) Relative, which analyzes on the attention map between the start and end of the given sentence.

The AttentionScore and probability-based baselines are white-box methods, requiring access to the model. Since we cannot access all models used to generate the datasets, we passed the sequence of tokens to the proxy LLM to obtain the scores. The proxy LLM was the same as the one used in our method.

As discussed in Section 2, fact-level methods like GraphEval and FactAlign require external knowledge source as reference, while GCA tuned hyperparameters on test data. These design differences make quantitative comparison with our zero-resource approach methodologically inappropriate.

[4] huggingface.co/datasets/potsawee/wiki_bio_gpt3_hallucination (cc-by-sa-3.0 license)

[5] huggingface.co/potsawee/deberta-v3-large-mnli (Apache license 2.0)

4.3 Implementation Details

We employed the Llama-3.1-70B-Instruct model (Grattafiori et al., 2024) as the LLM in all steps of our method and for baselines requiring access to LLM. We hosted it locally using vLLM (Kwon et al., 2023) on a server with 2 x Nvidia H100 94GB. For annotation of facts and hallucination correction, we utilized GPT-4o (OpenAI et al., 2024), as motivated in Sections 4.1 and 4.6. We set the LLM's temperature to 0.0 for all calls, except during hallucination correction (see Section 4.6), where we set it to 0.5. The used prompts are available in the code repository. We implemented the methods and experiments using LangChain (Chase, 2022), Hugging Face Transformers (Wolf et al., 2020) and Hugging Face Datasets (Lhoest et al., 2021). All pipeline steps were defined using DVC (Kuprieiev et al., 2025) to facilitate reproducibility.

To determine whether a language model's response was 'yes' or 'no' in Equation 9, we parsed the text into individual words and verified the presence of the words 'yes' or 'no'. If either word was detected, the response was excluded from the averaging process in Equations 8 and 10. Moreover, our pipeline is vulnerable to not detecting facts in short, uninformative sentences; for these cases, we set the score to 0.5. The analysis of the number of such sentences is provided in Appendix C.4.

4.4 Sentence-Level Detection

For a fair comparison, we employed the same evaluation protocol as SelfCheckGPT. We reported area under the precision-recall curve (AUC-PR). We ensured consistency in the evaluation protocol by reviewing their source code [6].

4.5 Fact-Level Detection

We evaluated all variants of FactSelfCheck using the fact-level score $\mathcal{H}_{\text{fact}}$. For comparison with SelfCheckGPT we used the best performing variant – Prompt. As SelfCheckGPT provides only sentence-level granularity, we derived fact-level scores by averaging the sentence-level scores across all sentences containing each fact.

4.6 Role of Fact-Level Detection in Hallucination Correction

One potential application of hallucination detection methods is their use in correcting hallucinated responses. In this experiment, we investigate the effectiveness of our fact-level detection approach in enhancing hallucination correction and compare the results with those obtained using sentence-level detection and a baseline method. Each of the three tested approaches uses different input for the LLM: **(1) Baseline**: the original prompt and the generated response. **(2) Sentence-level**: the original prompt, the generated response, and a list of hallucinated sentences. **(3) Fact-level**: the original prompt, the generated response, and a list of hallucinated facts.

As only the WikiBio dataset provides reference in form of the real Wikipedia biography, and the FavaMultiSamples dataset does not, we conducted this experiment only on WikiBio. The original prompt is the one used during the creation of the dataset: *"This is a Wikipedia passage about {concept_name}:"*. We instructed the LLM to return a list of sentences, allowing it to correct each sentence or leave it unchanged if no hallucinations were detected. We obtained the lists of incorrect sentences/facts using the best variants of models on this dataset (see Section 5.1) with thresholds that achieved the highest F1-scores on the dataset (0.3 for FactSelfCheck-Text and 0.75 for SelfCheckGPT (Prompt)).

Subsequently, we evaluated the factuality of the corrected responses using the LLM-as-judge approach (Zheng et al., 2023). For each corrected sentence, we provided the external biography from Wikipedia as a reference. As elaborated in Section 4.1, this approach has been tested for annotating hallucinations and demonstrates high alignment with human annotations (Janiak et al., 2025). We instructed LLM-as-judge to return 'yes' if the source supported the sentence, 'no' if it was not, or 'refused' if the LLM declined to correct the sentence (e.g., due to insufficient knowledge). We then categorized the responses into three labels: 'factual', 'non-factual', 'refused'.

As mentioned in Section 4.3, we utilized GPT-4o for correction and judging instead of Llama-3.1-70B-Instruct (used in detection). This choice was motivated by the challenging nature of the correction task – the model needs to correct hallucinations using only its internal knowledge, without access to external references. While the model knows hallucinated parts, it must rely on its knowledge to determine the correct information.

While the method of correction described here is not our main contribution, we used it to study the potential benefits of fact-level detection. Although the correction method employed here may not be

[6] github.com/potsawee/selfcheckgpt

the most sophisticated, the key takeaway is the observed difference in performance.

5 Results

This section presents the results of the experiments described in Section 4. The additional results are presented in Appendix C.

5.1 Sentence-Level Detection

Method	Agg.	AUC-PR
Sentence-level methods		
SCGPT (Prompt)	-	93.60
SCGPT (NLI)	-	92.50
AttentionScore (Relative)	-	83.85
Max($\mathcal{H}$)	-	82.56
Mean($-\log p$)	-	79.20
Mean($\mathcal{H}$)	-	79.02
Max($-\log p$)	-	78.41
AttentionScore (Absolute)	-	77.95
RandomSentence	-	72.96
Fact-level methods (ours)		
FSC-Text	max	92.45
FSC-KG (LLM-based)	max	91.82
FSC-Text	mean	91.01
FSC-KG (LLM-based)	mean	90.24
FSC-KG (Frequency-based)	max	88.48
FSC-KG (Frequency-based)	mean	88.25
RandomFact	mean	74.22

Table 1: WikiBio: Results on the sentence-level hallucination detection task. Comparison of sentence-level and fact-level methods based on AUC-PR scores. SCGPT stands for SelfCheckGPT, FSC represents FactSelfCheck, and Agg denotes the aggregation method used for calculating sentence-level scores.

Tables 1 and 2 present a comparative analysis of our method against baselines. For WikiBio, FactSelfCheck-Text utilizing *max* as an aggregation function achieves an AUC-PR score of 92.45. It demonstrates that our approach is comparable in performance to the leading SelfCheckGPT (SCGPT) variants – Prompt (93.60) and NLI (92.50). Notably, while our method operates at a more granular level, it maintains competitive performance with a marginal decrease of 1.2% compared to the best SCGPT. It is important to note that comparing our method to SelfCheckGPT at the sentence level inherently disadvantages our approach, as we operate at a more granular level of analysis. Nevertheless, our method still achieves competitive performance despite this inherent challenge.

Method	Agg.	AUC-PR
Sentence-level methods		
SCGPT (Prompt)	-	46.91
SCGPT (NLI)	-	32.58
Max($\mathcal{H}$)	-	28.22
Max($-\log p$)	-	26.20
AttentionScore (Relative)	-	24.17
Mean($\mathcal{H}$)	-	23.80
Mean($-\log p$)	-	22.85
AttentionScore (Absolute)	-	22.19
RandomSentence	-	21.70
Fact-level methods (ours)		
FSC-KG (Frequency-based)	max	48.52
FSC-Text	max	42.80
FSC-KG (LLM-based)	max	40.63
FSC-Text	mean	37.13
FSC-KG (Frequency-based)	mean	36.16
FSC-KG (LLM-based)	mean	35.81
RandomFact	mean	21.22

Table 2: FavaMultiSamples: Results on the sentence-level hallucination detection task. Comparison of sentence-level and fact-level methods based on AUC-PR scores. SCGPT stands for SelfCheckGPT, FSC represents FactSelfCheck, and Agg denotes the aggregation method used for calculating sentence-level scores.

For FavaMultiSamples, FactSelfCheck-KG (Frequency-based) with *max* aggregation achieves the highest AUC-PR score of 48.52, outperforming all sentence-level baselines, including SelfCheckGPT (Prompt) at 46.91. This result, together with the findings from WikiBio, highlights that the best-performing FactSelfCheck (FSC) variant depends on the dataset. On WikiBio, FSC-Text, which performs direct comparison between facts and samples, consistently achieves the highest hallucination AUC-PR (92.45), outperforming FSC-KG, which relies on knowledge graph comparisons. FSC-Text offers computational advantages by eliminating the knowledge graph extraction step. However, on FavaMultiSamples, the frequency-based FSC-KG variant surpasses both FSC-Text and LLM-based FSC-KG. These differences suggest that the optimal variant is influenced by dataset characteristics, such as fact density, sentence length, and text style (e.g., the prevalence of lists in FavaMultiSamples). FSC-KG may be more computationally efficient for longer samples with lower fact density due to reduced token usage, while FSC-Text is preferable for shorter or denser samples. Regarding aggregation functions, max outperformed $mean$ across both datasets. This makes intuitive sense – a sentence is hallucinated if any fact within it is hallucinated.

An interesting side observation is that our reproduced SCGPT (Prompt) with Llama-3.1-70B-Instruct marginally surpassed the original implementation using GPT-3.5-turbo (93.60 vs 93.42 [7]).

5.2 Fact-Level Detection

Method	AUC-PR
FactSelfCheck-Text	93.41
FactSelfCheck-KG (LLM-based)	92.25
FactSelfCheck-KG (Freq.-based)	87.99
SelfCheckGPT (Prompt)	86.18
RandomFact	65.79

Table 3: Results on the fact-level hallucination detection task. Comparison of sentence-level and fact-level methods based on AUC-PR scores.

Table 3 presents the comparative results for fact-level hallucination detection on WikiBio. FactSelfCheck-Text demonstrates superior performance with an AUC-PR score of 93.41, followed by FactSelfCheck-KG (LLM-based) at 92.25. The frequency-based method achieves 87.99, while SelfCheckGPT (Prompt) scores 86.18. These results demonstrate the effectiveness of our method in detecting hallucinations at the fact level. Furthermore, the lower performance of averaging the sentence-level scores highlights the importance of designing fact-level methods and validating our approach.

Comparing FactSelfCheck to SelfCheckGPT could be seen as unfair because the latter operates at a lower granularity than required by the evaluation task. However, this situation is analogous to the sentence-level evaluation presented in Section 5.1. In both cases, direct comparisons are not fully appropriate. The key difference is that, at the fact level, FactSelfCheck significantly outperforms SelfCheckGPT, while at the sentence level, FactSelfCheck remains competitive despite the unfavorable comparison.

5.3 Role of Fact-Level Detection in Hallucination Correction

Table 4 presents the results of our hallucination correction experiment. The fact-level approach shows substantial improvements over the baseline and sentence-level methods. We observed a 35.5% increase in factual content and 12.5% reduction in non-factual content compared to the baseline. In contrast, the sentence-level detection achieves only modest improvements of 10.6% and 4.8%, respectively, indicating that pointing out hallucinations at the fact level enables more effective corrections and underscoring the importance of our study and contributions.

The overall rate of refusals remains low, increasing only marginally from 0.04 in the baseline to 0.05 with fact-level and sentence-level detection. We hypothesize that the model becomes more cautious with provided information about hallucinations and more likely to know its limitations.

Level	Factual ↑	Non-Factual ↓	Refused
baseline	0.23	0.74	0.04
sentence	0.25 (+10.6%)	0.70 (-4.8%)	0.05 (+30.0%)
fact	0.31 (+35.5%)	0.64 (-12.5%)	0.05 (+30.0%)

Table 4: Effectiveness of hallucination correction by providing detected hallucinations at sentence-level, fact-level, and a baseline (without providing any hallucinations). The table presents the proportions of factual, non-factual sentences, and refused corrections. Percentages in parentheses indicate the relative change compared to the baseline. Arrows ↑ and ↓ denote whether a higher or lower value is better.

6 Conclusion

We introduced FactSelfCheck, a fact-level hallucination detection approach that achieves competitive performance with existing methods while providing more interpretable insights through structured knowledge representation. By detecting hallucinations at the granular fact level rather than sentence level, our method enables more effective hallucination correction through precise identification of incorrect facts. The zero-resource nature of our approach makes it broadly applicable across diverse domains without requiring external knowledge bases or domain-specific training data. Additionally, we contributed FavaMultiSamples, a novel benchmark addressing the critical gap in evaluation datasets for sampling-based hallucination detection methods.

[7] The scores on GPT-3.5-turbo were obtained from the original paper (Manakul et al., 2023).

Limitations

Our study faces three primary limitations. First, we are constrained by the availability of suitable datasets. Although we contributed FavaMultiSamples, the second dataset for evaluating sampling-based methods, it is still a dataset with annotations at the sentence-level. The lack of datasets with long generated passages and annotations at the fact-level forced us to evaluate our method through aggregation rather than directly assessing our fact-level detection capabilities. Second, while the granular approach of FactSelfCheck justifies its increased complexity, the multiple LLM-based steps make it more computationally intensive compared to more straightforward methods like SelfCheckGPT. Third, our method could face challenges with very short or uninformative sentences where fact extraction may fail.

Several promising directions could address these limitations. An important step would be creating new datasets with fact-level hallucination annotations, enabling direct evaluation of our method's core capabilities. Additionally, we see significant potential for improving computational efficiency. Our current prompt engineering was largely empirical and not optimized for token usage. Future work could focus on reducing prompt lengths and merging steps, such as merging the KG extraction steps or simultaneously assessing support for multiple facts.

Ethical Considerations

Like all machine learning methods, FactSelfCheck can produce false positives and false negatives. Therefore, it should not completely replace human verification of factual correctness in LLM responses.

Acknowledgments

This work was funded by the European Union under the Horizon Europe grant OMINO – Overcoming Multilevel INformation Overload (grant number 101086321, `http://ominoproject.eu/`). Views and opinions expressed are those of the authors alone and do not necessarily reflect those of the European Union or the European Research Executive Agency. Neither the European Union nor the European Research Executive Agency can be held responsible for them. It was also co-financed with funds from the Polish Ministry of Education and Science under the programme entitled International Co-Financed Projects, grant no. 573977. This work was co-funded by the National Science Centre, Poland under CHIST-ERA Open & Reusable Research Data & Software (grant number 2022/04/Y/ST6/00183). The infrastructure was provided by the CLARIN-PL project financed as part of the investment: "CLARIN ERIC – European Research Infrastructure Consortium: Common Language Resources and Technology Infrastructure (period: 2024-2026) funded by the Polish Ministry of Science and Higher Education (Programme: "Support for the participation of Polish scientific teams in international research infrastructure projects"), agreement number 2024/WK/01.

References

Amos Azaria and Tom Mitchell. 2023. The internal state of an LLM knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 967–976, Singapore. Association for Computational Linguistics.

Harrison Chase. 2022. Langchain.

Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. 2024. Inside: Llms' internal states retain the power of hallucination detection. *Preprint*, arXiv:2402.03744.

I-Chun Chern, Steffi Chern, Shiqi Chen, Weizhe Yuan, Kehua Feng, Chunting Zhou, Junxian He, Graham Neubig, and Pengfei Liu. 2023. Factool: Factuality detection in generative ai – a tool augmented framework for multi-task and multi-domain scenarios. *Preprint*, arXiv:2307.13528.

Yung-Sung Chuang, Linlu Qiu, Cheng-Yu Hsieh, Ranjay Krishna, Yoon Kim, and James R. Glass. 2024. Lookback lens: Detecting and mitigating contextual hallucinations in large language models using only attention maps. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1419–1436, Miami, Florida, USA. Association for Computational Linguistics.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2024. Chain-of-verification reduces hallucination in large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3563–3578, Bangkok, Thailand. Association for Computational Linguistics.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. *Preprint*, arXiv:2404.16130.

Xinyue Fang, Zhen Huang, Zhiliang Tian, Minghui Fang, Ziyi Pan, Quntian Fang, Zhihua Wen, Hengyue Pan, and Dongsheng Li. 2024. Zero-resource hallucination detection for text generation via graph-based contextual knowledge triples modeling. *arXiv preprint arXiv:2409.11283*.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. 2024. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630. Publisher: Nature Publishing Group.

Thomas Palmeira Ferraz, Kartik Mehta, Yu-Hsiang Lin, Haw-Shiuan Chang, Shereen Oraby, Sijia Liu, Vivek Subramanian, Tagyoung Chung, Mohit Bansal, and Nanyun Peng. 2024. LLM self-correction with DeCRIM: Decompose, critique, and refine for enhanced following of instructions with multiple constraints. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7773–7812, Miami, Florida, USA. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Representation learning on graphs: Methods and applications. In *IEEE Data Eng. Bull.*, volume 40, pages 52–74.

Xiangkun Hu, Dongyu Ru, Lin Qiu, Qipeng Guo, Tianhang Zhang, Yang Xu, Yun Luo, Pengfei Liu, Yue Zhang, and Zheng Zhang. 2024. Knowledge-centric hallucination detection. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6953–6975, Miami, Florida, USA. Association for Computational Linguistics.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Trans. Inf. Syst.*, 43(2).

Denis Janiak, Jakub Binkowski, Albert Sawczyn, Bogdan Gabrys, Ravid Shwartz-Ziv, and Tomasz Kajdanowicz. 2025. The illusion of progress: Re-evaluating hallucination detection in llms. *Preprint*, arXiv:2508.08285.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, and 17 others. 2022. Language models (mostly) know what they know. *Preprint*, arXiv:2207.05221.

Jannik Kossen, Jiatong Han, Muhammed Razzak, Lisa Schut, Shreshth Malik, and Yarin Gal. 2024. Semantic entropy probes: Robust and cheap hallucination detection in llms. *Preprint*, arXiv:2406.15927.

Ruslan Kuprieiev, skshetry, Peter Rowlands, Dmitry Petrov, Paweł Redzyński, Casper da Costa-Luis, David de la Iglesia Castro, Alexander Schepanovski, Ivan Shcheklein, Gao, Batuhan Taskaya, Jorge Orpinel, Fábio Santos, Daniele, Ronan Lamy, Aman Sharma, Zhanibek Kaimuldenov, Dani Hodovic, Nikita Kodenko, and 9 others. 2025. Dvc: Data version control - git for data & models.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Minhyeok Lee. 2023. A mathematical investigation of hallucination and creativity in gpt models. *Mathematics*, 11(10).

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, and 13 others. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Zicheng Lin, Zhibin Gou, Tian Liang, Ruilin Luo, Haowei Liu, and Yujiu Yang. 2024. CriticBench: Benchmarking LLMs for critique-correct reasoning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1552–1587, Bangkok, Thailand. Association for Computational Linguistics.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017, Singapore. Association for Computational Linguistics.

Ning Miao, Yee Whye Teh, and Tom Rainforth. 2023. Selfcheck: Using llms to zero-shot check their own step-by-step reasoning. *Preprint*, arXiv:2308.00436.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.

Abhika Mishra, Akari Asai, Vidhisha Balachandran, Yizhong Wang, Graham Neubig, Yulia Tsvetkov, and Hannaneh Hajishirzi. 2024. Fine-grained hallucinations detections. *arXiv preprint*.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, and 401 others. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Liangming Pan, Michael Saxon, Wenda Xu, Deepak Nathani, Xinyi Wang, and William Yang Wang. 2024. Automatically correcting large language models: Surveying the landscape of diverse automated correction strategies. *Transactions of the Association for Computational Linguistics*, 12:484–506.

Mohamed Rashad, Ahmed Zahran, Abanoub Amin, Amr Abdelaal, and Mohamed Altantawy. 2024. FactAlign: Fact-level hallucination detection and classification through knowledge graph alignment. In *Proceedings of the 4th Workshop on Trustworthy Natural Language Processing (TrustNLP 2024)*, pages 79–84, Mexico City, Mexico. Association for Computational Linguistics.

Malik Sallam. 2023. Chatgpt utility in healthcare education, research, and practice: Systematic review on the promising perspectives and valid concerns. *Healthcare*, 11(6).

Hannah Sansford, Nicholas Richardson, Hermina Petric Maretic, and Juba Nait Saada. 2024. Grapheval: A knowledge-graph based llm hallucination evaluation framework. *Preprint*, arXiv:2407.10793.

Gaurang Sriramanan, Siddhant Bharti, Vinu Sankar Sadasivan, Shoumik Saha, Priyatham Kattakinda, and Soheil Feizi. 2024. LLM-check: Investigating detection of hallucinations in large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Aman Singh Thakur, Kartik Choudhary, Venkat Srinik Ramayapally, Sankaran Vaidyanathan, and Dieuwke Hupkes. 2025. Judging the judges: Evaluating alignment and vulnerabilities in llms-as-judges. *Preprint*, arXiv:2406.12624.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. 2024. Hallucination is inevitable: An innate limitation of large language models. *Preprint*, arXiv:2401.11817.

Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A. Smith. 2024a. How language model hallucinations can snowball. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Xiaoying Zhang, Baolin Peng, Ye Tian, Jingyan Zhou, Lifeng Jin, Linfeng Song, Haitao Mi, and Helen Meng. 2024b. Self-alignment for factuality: Mitigating hallucinations in LLMs via self-evaluation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1965, Bangkok, Thailand. Association for Computational Linguistics.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2023. Siren's song in the ai ocean: A survey on hallucination in large language models. *Preprint*, arXiv:2309.01219.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Appendix

This appendix provides supplementary material organized as follows:

Appendix A: FavaMultiSamples dataset creation and annotation methodology.

Appendix B: Comprehensive statistics for both evaluation datasets.

Appendix C: Additional results including complete sentence-level metrics with precision-recall curves, sample size ablation study, and evaluation of the intermediate steps of the pipeline. As well as confusion matrices and prediction statistics for the fact-level evaluation.

Appendix D: Computational complexity analysis.

Appendix E: Concrete example with knowledge graph visualization demonstrating fact-level detection advantages.

Appendix F: Enhanced prompt experiment validating comparison fairness with SelfCheckGPT.

A FavaMultiSamples Dataset

Addressing the lack of evaluation benchmarks for sampling-based hallucination detection methods, we built FavaMultiSamples upon the FAVA dataset[8] developed by Mishra et al. (2024). Prior to our contribution, WikiBio was the only available dataset for evaluating sampling-based methods, limiting evaluation of different approaches. The original FAVA dataset contains 460 passages generated by GPT (gpt-3.5-turbo-0301) and Llama2-Chat-70B in response to diverse information-seeking prompts. Each passage was annotated by trained annotators for factual accuracy. For more details about the dataset construction and annotation process, please refer to the original paper.

To create FavaMultiSamples, we generated 20 samples for each passage with the temperature of 1.0, matching the sample settings used in the WikiBio dataset. We used the same models that produced the original responses, with one exception: since gpt-3.5-turbo-0301 is no longer available, we used gpt-3.5-turbo-1106, the most similar model available at the time. The FAVA dataset uses an HTML-like format for annotations, so we split each generated response into sentences and annotated them in binary format, where 1 indicates a sentence containing a hallucination.

B Dataset Statistics

The statistics of the used datasets are presented in Table 5.

	WikiBio	**FavaMulti-Samples**
# Passages	238	460
# Sentences	1908	5660
# Hall. sentences	1392	1228
# Fact. sentences	516	4432
% Hall. sentences	72.96	21.70
Avg. sent./passage	8.02	12.30
Avg. tok./passage	184.77	340.72
Avg. tok./sentence	23.48	30.30

Table 5: Statistics of the used datasets: WikiBio and FavaMultiSamples. We summarize the number of passages, sentences, tokens and annotation statistics. The number of tokens were calculated using appropriate tokenizers for each model.

C Additional Results

C.1 Sentence-Level Detection

Tables 6 and 7 provide a more comprehensive view of the FactSelfCheck variants against baselines on both datasets, including AUC-PR scores for detecting factual sentences (Factuality AUC-PR) in addition to hallucinated ones, and an average of these two. While Section 5.1 focused on hallucination detection, these extended results offer additional insights into our approach. On WikiBio, FSC-Text (max aggregation) shows comparable performance in hallucination detection with 92.45 AUC-PR vs. 93.60 for SCGPT (Prompt). While SCGPT (Prompt) achieves a higher factuality detection score (74.30 vs. 65.55), our method performs well in identifying potential misinformation and contributes to effective factual verification.

On the FavaMultiSamples dataset, FSC-KG (Frequency-based, max aggregation) demonstrates high overall performance, achieving a factuality detection AUC-PR of 80.21 alongside solid hallucination detection (48.52 AUC-PR compared to SCGPT Prompt's 46.91). This performance across metrics shows the adaptability of our approach to different datasets. While SCGPT (Prompt) achieves a higher average AUC-PR (67.65 vs. 64.36) due to stronger factuality detection (88.39), our method provides balanced performance across

[8] https://huggingface.co/datasets/fava-uw/fava-data license: CC-BY-4.0

Method	Agg.	AUC-PR		
		Hallucination	Factuality	Avg.
Sentence-level methods				
SCGPT (Prompt)	-	93.60	74.30	83.95
SCGPT (NLI)	-	92.50	66.08	79.29
AttentionScore (Relative)	-	83.85	51.62	67.74
Max($\mathcal{H}$)	-	82.56	41.80	62.18
Mean($-\log p$)	-	79.20	44.11	61.65
Mean($\mathcal{H}$)	-	79.02	49.18	64.10
Max($-\log p$)	-	78.41	33.59	56.00
AttentionScore (Absolute)	-	77.95	42.23	60.09
RandomSentence	-	72.96	27.04	50.00
Fact-level methods (ours)				
FSC-Text	max	92.45	65.55	79.00
FSC-KG (LLM-based)	max	91.82	64.64	78.23
FSC-Text	mean	91.01	63.77	77.39
FSC-KG (LLM-based)	mean	90.24	62.95	76.60
FSC-KG (Frequency-based)	max	88.48	53.86	71.17
FSC-KG (Frequency-based)	mean	88.25	55.27	71.76
RandomFact	mean	74.22	29.74	51.98

Table 6: WikiBio: Extended results on the sentence-level hallucination detection task. Comparison of sentence-level and fact-level methods based on AUC-PR scores for Hallucination, Factuality, and their Average. SCGPT stands for SelfCheckGPT, FSC represents FactSelfCheck, and Agg denotes the aggregation method used for calculating sentence-level scores. The results are sorted by AUC-PR scores for Hallucination.

both metrics, offering advantages for applications where hallucination detection is important.

Figures 2 and 3 illustrate the precision-recall curves for the sentence-level detection task on the WikiBio and FavaMultiSamples datasets, respectively. These curves show the performance across various thresholds for both hallucination and factuality detection.

C.2 Fact-Level Detection

To extend the demonstration of the effectiveness of our fact-level detection approach, we present confusion matrices and prediction statistics. Table 8 shows the confusion matrix for FactSelfCheck-Text, while Table 9 presents results for SelfCheckGPT (Prompt) when adapted to fact-level prediction.

FactSelfCheck-Text correctly identifies 3530 hallucinated facts and 951 factual facts, achieving higher true positive rates for hallucination detection compared to SelfCheckGPT, which correctly identifies 3503 hallucinated facts but only 570 factual facts.

Table 10 summarizes the overall prediction accuracy of both methods. FactSelfCheck achieves 4481 correct predictions compared to 4073 for SelfCheckGPT, representing an increase of 408 correct predictions (10.02% improvement).

C.3 Effect of Sample Size on Detection Performance

We investigated the impact of varying the number of samples on the detection performance of our method. Specifically, we evaluated the performance at the sentence level by changing the number of samples from 1 to 20. We compared our method with SelfCheckGPT.

Figure 4 illustrates that that on WikiBio, FactSelfCheck exhibits similar behavior to SelfCheckGPT regarding sample requirements. The performance improves dramatically with up to 5 samples, after which the improvement curve flattens. While additional samples continue to yield benefits, these improvements become incremental, with modest gains observed up to 20 samples. Factuality detection exhibits similar patterns to hallucination detection. This pattern confirms that more samples provide better evidence for accurate detection

Method	Agg.	AUC-PR		
		Hallucination	Factuality	Avg.
Sentence-level methods				
SCGPT (Prompt)	-	46.91	88.39	67.65
SCGPT (NLI)	-	32.58	85.64	59.11
Max($\mathcal{H}$)	-	28.22	81.90	55.06
Max($-\log p$)	-	26.20	80.84	53.52
AttentionScore (Relative)	-	24.17	82.12	53.15
Mean($\mathcal{H}$)	-	23.80	80.62	52.21
Mean($-\log p$)	-	22.85	80.38	51.62
AttentionScore (Absolute)	-	22.19	81.71	51.95
RandomSentence	-	21.70	78.30	50.00
Fact-level methods (ours)				
FSC-KG (Frequency-based)	max	48.52	80.21	64.36
FSC-Text	max	42.80	86.15	64.47
FSC-KG (LLM-based)	max	40.63	85.01	62.82
FSC-Text	mean	37.13	86.09	61.61
FSC-KG (Frequency-based)	mean	36.16	79.92	58.04
FSC-KG (LLM-based)	mean	35.81	84.65	60.23
RandomFact	mean	21.22	77.77	49.50

Table 7: FavaMultiSamples: Extended results on the sentence-level hallucination detection task. Comparison of sentence-level and fact-level methods based on AUC-PR scores for Hallucination, Factuality, and their Average. SCGPT stands for SelfCheckGPT, FSC represents FactSelfCheck, and Agg denotes the aggregation method used for calculating sentence-level scores. The results are sorted by AUC-PR scores for Hallucination.

	Pred Fact.	Pred Hall.
True Fact.	951	832
True Hall.	175	3530

Table 8: Confusion matrix of FactSelfCheck-Text on fact-level evaluation.

	Pred Fact.	Pred Hall.
True Fact.	570	1213
True Hall.	202	3503

Table 9: Confusion matrix of SelfCheckGPT (Prompt) on fact-level evaluation.

Method	Correct	Incorrect
SelfCheckGPT	4073	1415
FactSelfCheck	4481	1007

Table 10: Overall prediction accuracy comparison for fact-level evaluation.

across both metrics.

Figure 5 shows a different pattern for FavaMultiSamples, where hallucination detection performance decreases with more samples. This occurs because methods with few samples tend to overestimate hallucination scores, interpreting normal variations as potential hallucinations. As sample size increases, methods become more conservative in their scoring, leading to better calibration. This pattern likely stems from FavaMultiSamples having shorter sentences with fewer facts compared to WikiBio (see Section C.4).

The distinct patterns between datasets highlight that sampling effects are context-dependent. While more samples universally improve calibration quality, the impact on raw performance metrics depends on dataset characteristics and initial score distributions. It's important to note that AUC-PR, while useful, has limitations. A classifier that consistently returns the same score can achieve a high AUC-PR value, which may not reflect true discriminative ability. Therefore, the decrease in AUC-PR with more samples might actually indicate better calibration and more meaningful score distributions, rather than worse performance.

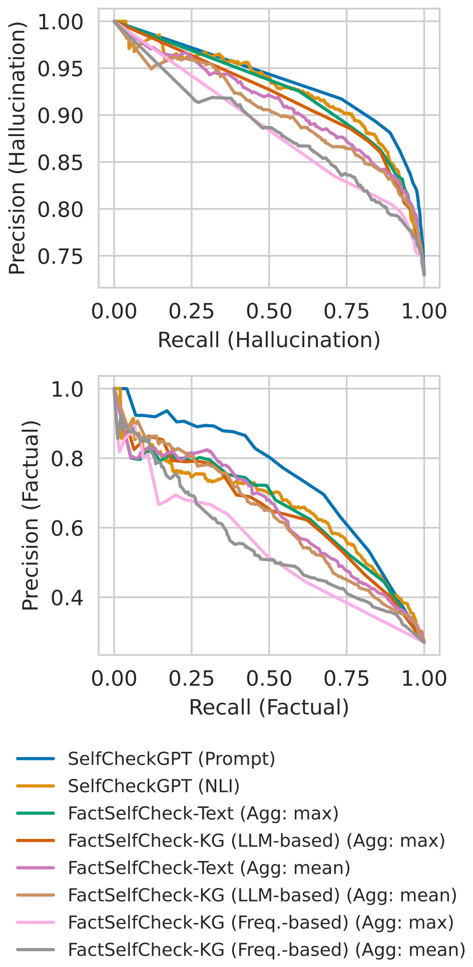

Figure 2: WikiBio: Precision-recall curve for the sentence-level hallucination and factuality detection.

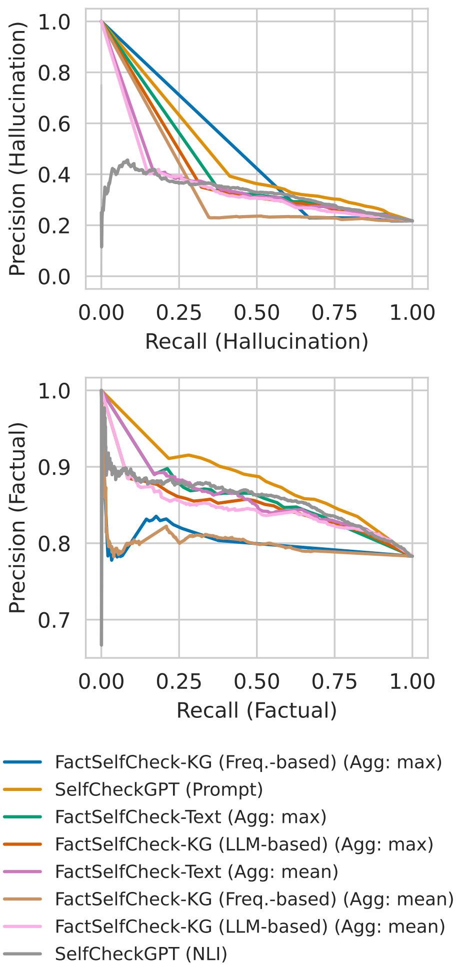

Figure 3: FavaMultiSamples: Precision-recall curve for the sentence-level hallucination and factuality detection.

C.4 Evaluation of Intermediate Steps

Our method consists of multiple steps that cannot be directly evaluated due to the lack of human annotations. While previous sections evaluated the complete pipeline, to strengthen our study we also analyzed and validated statistics from intermediate steps. We examined the number of entities and relations per passage, along with facts per sentence, calculating mean, minimum, maximum values, and the count and percentage of entries with zero elements. Sentences with no facts are particularly important as FactSelfCheck assigns them a default score of 0.5 (see Section 4.3).

Table 11 reveals that both datasets have high mean numbers of entities and relations per passage, indicating that knowledge graph construction is not constrained by earlier steps. While WikiBio shows acceptable minimum values for entities and relations per passage, FavaMultiSamples exhibits notably lower minimums that could impact knowledge graph extraction performance. The percentage of sentences without extracted facts is relatively low in WikiBio (1.15%) but more substantial in FavaMultiSamples (6.08%), potentially affecting detection accuracy. Both datasets show considerable variability, with some passages containing over 200 entities and relations, highlighting the diverse complexity of the analyzed generated responses.

D Computational complexity

While FactSelfCheck is more granular than SelfCheckGPT, this comes with increased computational costs. In Table 12 we compare all variants of FactSelfCheck with SelfCheckGPT (Prompt) in terms of the number of LLM calls required. Every variant of FactSelfCheck requires additional calls for entity and relation extraction (constant 2), followed by knowledge graph construction for each sentence ($|U|$), then it assesses factual consis-

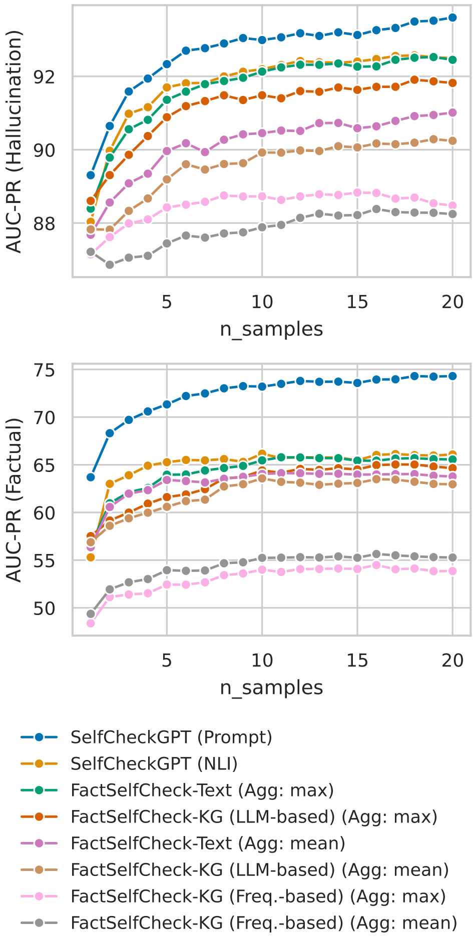

Figure 4: WikiBio: Impact of sample size on both hallucination and factuality detection performance for different methods.

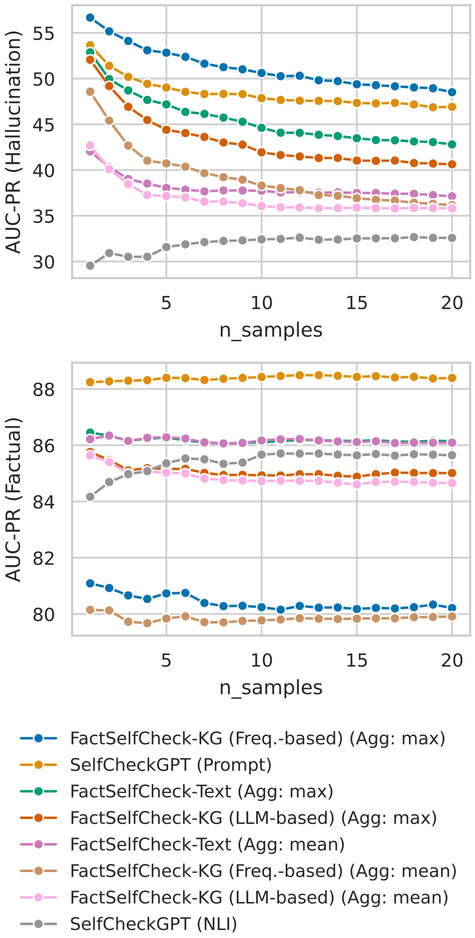

Figure 5: FavaMultiSamples: Impact of sample size on both hallucination and factuality detection performance for different methods.

tency of each fact in different ways with varying complexity. As noted in Limitations, we did not optimize for token usage, and future work could merge steps to reduce complexity while maintaining the fine-grained insights our method provides.

E Case Study of FactSelfCheck vs SelfCheckGPT

Table 13 and Figure 6 present a comparative case study of predictions from WikiBio made by FactSelfCheck and SelfCheckGPT. The table contains an external Wikipedia biography, sentences from the response, facts extracted from the response, and the predictions of both methods. This comparison demonstrates that fact-level detection provides more detailed information about the factuality of the response. We observe that LLM did not hallucinate all facts, as some are consistent with the external Wikipedia biography. However, when using sentence-level detection, we cannot distinguish between correct and hallucinated facts – all sentences are predicted as hallucinated.

F SelfCheckGPT with Enhanced Prompt

To ensure a fair comparison between FactSelfCheck and SelfCheckGPT, during preliminary studies, we conducted an additional experiment using an enhanced prompt for SelfCheckGPT. The original SelfCheckGPT prompt is relatively simple, while our FactSelfCheck prompts are more elaborate, directly allowing reasoning and inference of new facts, and providing examples. These characteristics could potentially increase the performance of methods. We designed an alternative prompt for SelfCheckGPT, that incorporates these features, making it similar to our FactSelfCheck prompts.

For sentence-level detection on WikiBio, the enhanced prompt for SelfCheckGPT achieved an AUC-PR score of 93.38, slightly lower than the

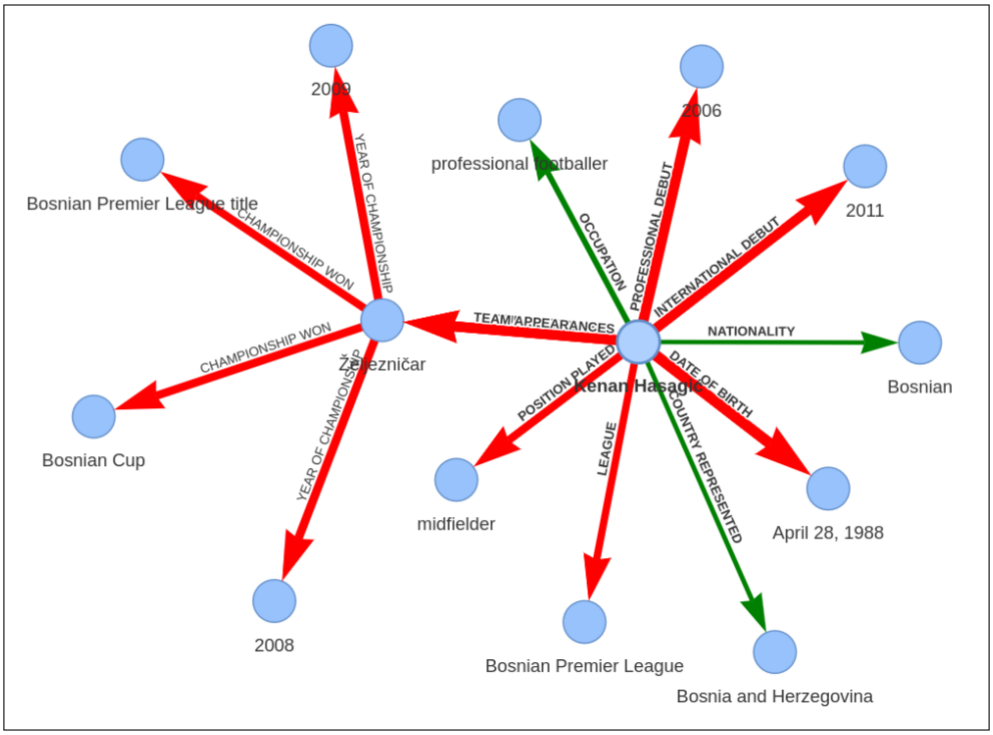

Figure 6: Example of a knowledge graph extracted from a response. Edge width represents the hallucination score for each fact, with red edges indicating hallucinated facts and green edges indicating correct facts. Facts were classified using a threshold of 0.3, which achieved the highest F1-score in the fact-level evaluation (C.2).

	# entities/ passage	**# relations/ passage**	**# facts/ sentence**
		WikiBio	
mean	24.85	21.18	3.24
min	11	5	0
max	282	163	85
# 0 el.	0	0	22
% 0 el.	0.00%	0.00%	1.15%
		FavaMultiSamples	
mean	32.08	50.34	2.97
min	1	0	0
max	312	237	102
# 0 el.	0	2	344
% 0 el.	0.00%	0.43%	6.08%

Table 11: Statistics of intermediate steps in FactSelfCheck across both datasets. The table shows the distribution of entities per passage, relations per passage, and facts per sentence, including occurrences of entries with 0 elements.

Method	**Number of LLM calls**
FSC-KG (Freq.)	$2 + \lvert U\rvert + \lvert S\rvert$
FSC-KG (LLM)	$2 + \lvert U\rvert + \lvert S\rvert + \lvert KG_p\rvert \times \lvert S\rvert$
FSC-Text	$2 + \lvert U\rvert + \lvert KG_p\rvert \times \lvert S\rvert$
SCGPT-Prompt	$\lvert U\rvert \times \lvert S\rvert$

Table 12: Computational complexity of different methods in terms of number of LLM calls required. U is the set of sentences in the generated passage, S is the set of stochastic LLM response samples, and KG_p is the knowledge graph containing all extracted facts.

original prompt's 93.60. This minimal difference indicates that SelfCheckGPT's performance is not significantly affected by prompt design in our experimental setting. In fact, the enhanced prompt even lowered the performance rather than increasing it, despite its more sophisticated design.

Due to these findings, we chose to use the original prompt for SelfCheckGPT in our experiments. These consistent results confirm that our comparison between methods is fair, as the enhanced prompt did not improve the performance of SelfCheckGPT.

<table>
<tr><th colspan="2">External Wikipedia Bio</th></tr>
<tr><td colspan="2">Kenan Hasagić (born 1 February 1980) is a Bosnian football goalkeeper who plays for Balıkesirspor. His football career began in his hometown with FK Rudar. At the age of 16, he made his debut in a first division match. He was the most promising goalkeeper in Bosnia and Herzegovina; he played for youth selections and was later transferred to Austrian side Vorwärts Steyr. After that, he was a member of Altay SK in Turkey but didn't see much first team football. He went back to Bosnia and played for Bosna Visoko. In 2003, he signed a contract with FK Željezničar. Here he found good form and even became first choice goalkeeper for the Bosnian national team. In the 2004–05 season, he moved to Turkey once again where he signed for Turkish Süper Lig side Gaziantepspor. He made his debut for the national team on 12 February 2003 in a game between Wales and Bosnia and Herzegovina which ended in a 2–2 draw.</td></tr>
<tr><th colspan="2"><b>Sentence 1</b> (SelfCheckGPT: 1.0)</th></tr>
<tr><td colspan="2">Kenan Hasagić (born 28 April 1988) is a Bosnian professional footballer who plays as a midfielder for Bosnian Premier League club Željezničar.</td></tr>
<tr><th>Fact</th><th>FactSelfCheck</th></tr>
<tr><td>('Kenan Hasagić', 'DATE OF BIRTH', 'April 28, 1988')</td><td>1.00</td></tr>
<tr><td>('Kenan Hasagić', 'NATIONALITY', 'Bosnian')</td><td>0.00</td></tr>
<tr><td>('Kenan Hasagić', 'OCCUPATION', 'professional footballer')</td><td>0.20</td></tr>
<tr><td>('Kenan Hasagić', 'POSITION PLAYED', 'midfielder')</td><td>0.55</td></tr>
<tr><td>('Kenan Hasagić', 'LEAGUE', 'Bosnian Premier League')</td><td>0.45</td></tr>
<tr><td>('Kenan Hasagić', 'CURRENT CLUB', 'Željezničar')</td><td>0.85</td></tr>
<tr><th colspan="2"><b>Sentence 2</b> (SelfCheckGPT: 1.0)</th></tr>
<tr><td colspan="2">Hasagić started his career at his hometown club Željezničar, where he made his professional debut in 2006.</td></tr>
<tr><th>Fact</th><th>FactSelfCheck</th></tr>
<tr><td>('Kenan Hasagić', 'CURRENT CLUB', 'Željezničar')</td><td>0.85</td></tr>
<tr><td>('Kenan Hasagić', 'PROFESSIONAL DEBUT', '2006')</td><td>0.90</td></tr>
<tr><td>('Kenan Hasagić', 'CURRENT CLUB', 'Željezničar')</td><td>0.85</td></tr>
<tr><th colspan="2"><b>Sentence 3</b> (SelfCheckGPT: 1.0)</th></tr>
<tr><td colspan="2">He has since gone on to make over 200 appearances for the club, winning the Bosnian Premier League title in 2008 and the Bosnian Cup in 2009.</td></tr>
<tr><th>Fact</th><th>FactSelfCheck</th></tr>
<tr><td>('Kenan Hasagić', 'TEAM APPEARANCES', 'Željezničar')</td><td>0.60</td></tr>
<tr><td>('Željezničar', 'CHAMPIONSHIP WON', 'Bosnian Premier League title')</td><td>0.90</td></tr>
<tr><td>('Željezničar', 'YEAR OF CHAMPIONSHIP', '2008')</td><td>1.00</td></tr>
<tr><td>('Željezničar', 'CHAMPIONSHIP WON', 'Bosnian Cup')</td><td>0.95</td></tr>
<tr><td>('Željezničar', 'YEAR OF CHAMPIONSHIP', '2009')</td><td>1.00</td></tr>
<tr><th colspan="2"><b>Sentence 4</b> (SelfCheckGPT: 0.9)</th></tr>
<tr><td colspan="2">He has also represented Bosnia and Herzegovina at international level, making his debut in 2011.</td></tr>
<tr><th>Fact</th><th>FactSelfCheck</th></tr>
<tr><td>('Kenan Hasagić', 'COUNTRY REPRESENTED', 'Bosnia and Herzegovina')</td><td>0.05</td></tr>
<tr><td>('Kenan Hasagić', 'INTERNATIONAL DEBUT', '2011')</td><td>0.85</td></tr>
</table>

Table 13: Comparison of fact-level FactSelfCheck with sentence-level SelfCheckGPT. An external Wikipedia biography is provided to analyse the correctness of the methods. The red value indicates hallucinations, and the green value indicates factual correctness. The facts were classified using a threshold of 0.3 utilizing FactSelfCheck, and the sentences were classified using a threshold of 0.75 with SelfCheckGPT. These thresholds achieved the highest F1-scores in fact-level (Appendix C.2) and sentence-level (Section 5.1) evaluation, respectively.