# G2LDetect: A Global-to-Local Approach for Hallucination Detection

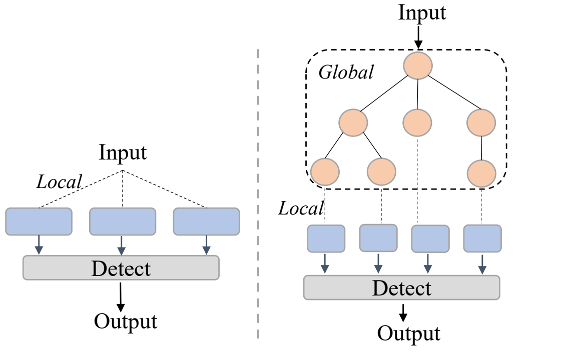
**Xiaoxia Cheng, Zeqi Tan, Zhe Zheng, Weiming Lu***

College of Computer Science and Technology, Zhejiang University
{zjucxx, zqtan, zhezheng, luwm}@zju.edu.cn

## Abstract

Hallucination detection has attracted considerable interest due to the tendency of language models to generate texts that contain hallucinations. Most existing methods start with specific local details directly extracted from text, then aggregate to form the final conclusion. However, this direct extraction approach ignores the global context, leading to isolated details, and is prone to missed or over-detections. In this paper, we present a global-to-local approach for hallucination detection (**G2LDetect**), which considers the global information of the text before identifying local details. We first construct a global representation of the text by transforming it into a hierarchical tree structure. Afterward, we obtain specific local details from the global tree representation using path-wise identification and perform detection on them. This global-to-local detection process ensures that local details are context-aware and complete, thus making more accurate and reliable detection results. Experimental results show that our global-to-local method outperforms existing methods, especially for longer texts.

**Code** — https://github.com/hustcxx/G2LDetect



(a) Existing (b) Global-to-Local

Figure 1: G2LDetect (Ours) vs. existing methods. (a) Existing methods start with specific local details directly extracted from text, then detect and aggregate to reach a final answer. (b) Our G2LDetect method constructs a global representation before extracting local details.

## 1 Introduction

Generative language models (OpenAI 2024; Touvron et al. 2023; Brown et al. 2020) have demonstrated their capabilities across various tasks, yet their generated content also faces the challenge of hallucinations (Ji et al. 2023; Bang et al. 2023; Qiu et al. 2023), which limits their application in real-world scenarios. In light of this, developing effective hallucination detection mechanisms is particularly important. Such mechanisms (Hu et al. 2024; Manakul, Liusie, and Gales 2023; Min et al. 2023; Chern et al. 2023) can evaluate and identify inaccurate or false information in the text against a reference, thereby enhancing authenticity and reliability of the language models generated content.

Recently, various hallucination detection methods (Hu et al. 2024; Manakul, Liusie, and Gales 2023; Min et al. 2023; Chern et al. 2023) have been proposed. These methods typically begin by extracting specific local details directly from the text, then perform separate detection, and finally aggregate the results to reach a conclusion, as shown in Figure 1 (a). For example, SelfCheckGPT (Manakul, Liusie, and Gales 2023) splits the text into sentences according to grammatical rules to perform independent detection, and then aggregate the results at the sentence level. FActScore (Min et al. 2023) further decomposes the sentences into sub-sentences using the in-context learning ability of large language models (LLMs) after rule-based sentence splitting. Subsequently, FacTool (Chern et al. 2023) introduces a method to directly extract sub-sentences employing the in-context learning ability of LLMs. Similarly, REFCHECKER (Hu et al. 2024) extracts triples from text to provide more fine-grained detection results than the sentence-level (Manakul, Liusie, and Gales 2023) and sub-sentence level (Min et al. 2023; Chern et al. 2023) approaches. However, these approaches directly extract factual units such as sentences, sub-sentences, and triples from the original documents without conducting a global semantic analysis of the entire document. This results in isolated factual units that lack context. The issue is particularly evident in rule-based approaches like SelfCheckGPT and FActScore. Moreover, sentence-level detection methods also suffer from the missed detection problem, since one sentence may contain multiple hallucinations. Additionally, according to our statistics, exist-

*Corresponding author.

ing methods such as FactScore extract an average of 4 factual units per sentence in the HaluEval-Summary dataset (Li et al. 2023). Based on the Cognitive Load Theory (Orru and Longo 2018), humans can only process 1 to 3 factual units in a single sentence, thus this approach may lead to the issue of over-detection.

To address the above issues, we propose a global-to-local hallucination detection method (**G2LDetect**). As shown in Figure 1(b), our method first constructs a global representation of the text by transforming it into a hierarchical tree structure utilizing LLMs. The global representation encompasses text elements and their relationships, providing a complete and structured view. Furthermore, it's tree structure can cluster related elements together, thereby avoiding discrete elements. Afterward, we extract local details from the global representation using a path-wise identification algorithm. The local details obtained through this method include all the information from the root node to the leaf nodes, comprising the context of the local details, thus enabling independent detection. Additionally, the path-wise identification method covers all details in the global representation, preventing missed or over-detections and allowing for fine-grained hallucination detection. Our proposed G2LDetect approach effectively maintains the integrity of both global context and semantics while ensuring the thoroughness of local details, thereby providing a robust solution for hallucination detection in complex texts. Experimental results on hallucination benchmarks, including HaluEval (Li et al. 2023), TRUE (Honovich et al. 2022), and datasets across different domains show that our approach outperforms previous methods, especially for longer texts.

Our main contributions are as follows:

- We propose a global-to-local approach for hallucination detection. This method first constructs a global representation by converting the text into a hierarchical tree structure before extracting local details. The representation clusters related elements together, thereby avoiding the presence of discrete elements.
- To obtain specific local details, we use a path-wise identification algorithm on the global representation. The identification method covers all aspects of global representation, preventing missed and over-detections and enabling fine-grained hallucination detection.
- Experiments improvements on HaluEval and TRUE benchmarks and other domain datasets demonstrate the effectiveness of our proposed method, especially for longer texts.

## 2 Related Work

Hallucination detection focuses on detecting the factuality of a text against the given references. Traditional hallucination detection methods generally rely on existing natural language inference (NLI) (Poliak 2020) or question answering (QA) methods (Zhang et al. 2023). With the development of LLMs (OpenAI 2024; Touvron et al. 2023; Brown et al. 2020), many hallucination detection methods based on LLMs have been proposed. Some methods (Chen et al. 2024; Su et al. 2024) use the internal state of large models for hallucination detection. However, these methods are only applicable when the internal state can be accessed. The more general approaches based on LLMs typically follow the method of decomposition, detection, and integration to arrive at the final conclusion. For example, SelfCheckGPT (Manakul, Liusie, and Gales 2023) splits the text into sentences, then uses the self-consistency ability of LLMs to detect hallucinations in each sentence, and finally aggregates the detection results. FActScore (Min et al. 2023) and FacTool (Chern et al. 2023) decompose the text into subsentences employing the in-context learning ability (Brown et al. 2020) of LLMs, then perform subsequent detection. REFCHEKER (Hu et al. 2024) extracts triples from the text, achieving more fine-grained hallucination detection. Different from the above methods starting with local details, we propose a global-to-local hallucination detection method, which first performs a global representation of the text and then gets the local details from the global representation.

## 3 Methods

In this section, we first introduce the task formulation of hallucination detection in §3.1 and then demonstrate each component of our method in detail. As shown in Figure 2, our method consists of three components: global representation §3.2, local details identification §3.3, and detection §3.4.

### 3.1 Task Formulation

Given a sample $(\mathcal{C}, \mathcal{D}, \mathcal{Y})$, where $\mathcal{C}$ and $\mathcal{D}$ represent the text to be checked and reference documents, respectively, the task is to predict a label $\mathcal{Y}$ to assess whether the text is TRUE or FALSE, grounded in the reference documents $\mathcal{D}$.

### 3.2 Global Representation

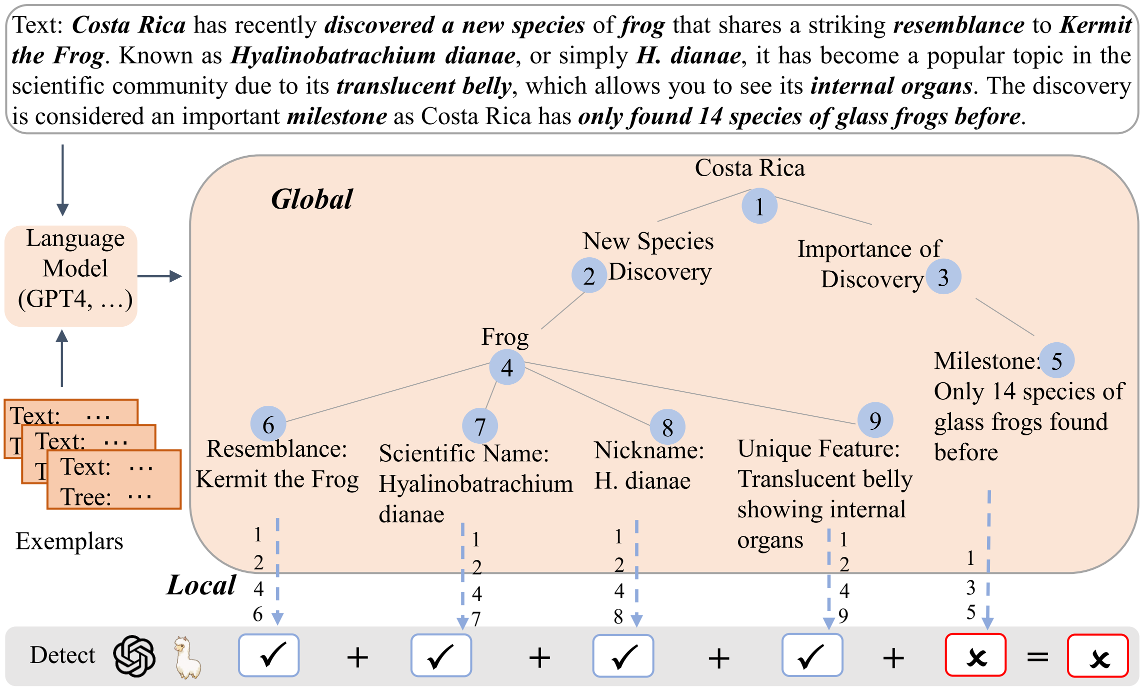
Our method starts with a global representation of the text before identifying the local details, which is the most apparent difference from existing approaches (Manakul, Liusie, and Gales 2023; Min et al. 2023; Chern et al. 2023; Hu et al. 2024). This global perspective mimics human cognitive strategies by using an initial broad understanding to guide detailed analysis. To effectively represent the global perspective of the text, we transform it into a hierarchical tree structure. The structure definition is as follows:

**Definition 1** (**Global Tree**). A Global Tree of text is a hierarchy structure representation of information, where the central theme or core idea is placed at the root, and various main concepts extend outward as branches. These main branches represent the major categories or ideas connected to the central theme. Each branch can further split into subbranches, which detail relevant facts, secondary concepts, or more granular pieces of information.

We leverage a pre-trained LLM to generate a global representation $\mathcal{G}$ of the text. The process can be defined as:

$$\mathcal{G} = f_{\theta}(\mathcal{C}, T_{in}) \tag{1}$$

where $f_{\theta}(\cdot)$ is a language model parameterized by $\theta$, $T_{in}$ is the task prompt. The task prompt $T_{in}$ consists of instructions



Text: ***Costa Rica*** has recently ***discovered a new species*** of ***frog*** that shares a striking ***resemblance*** to ***Kermit the Frog***. Known as ***Hyalinobatrachium dianae***, or simply ***H. dianae***, it has become a popular topic in the scientific community due to its ***translucent belly***, which allows you to see its ***internal organs***. The discovery is considered an important ***milestone*** as Costa Rica has ***only found 14 species of glass frogs before***.

Language Model (GPT4, …)
Text: ⋯
Text: ⋯
Text: ⋯
Tree: ⋯
Exemplars
Global
Costa Rica 1
New Species Discovery 2
Importance of Discovery 3
Frog 4
Milestone: 5 Only 14 species of glass frogs found before
6 Resemblance: Kermit the Frog
7 Scientific Name: Hyalinobatrachium dianae
8 Nickname: H. dianae
9 Unique Feature: Translucent belly showing internal organs
Local
1 2 4 6
1 2 4 7
1 2 4 8
1 2 4 9
1 3 5
Detect
✓ + ✓ + ✓ + ✓ + ✗ = ✗

Figure 2: Illustration of our global-to-local method for hallucination detection (**G2LDetect**). The global representation encompasses elements of the text and their relationships, providing a complete and structured view. Local details are obtained by using a path-wise identification algorithm. Then our method detects these local details and aggregates the detection results to reach a final conclusion.

and a definition of the hierarchical global tree structure. The specific prompt used in our paper is displayed as follows:

```
Global Tree Definition: [Definition 1]
Given an input text, please transform
the text into a global tree structure.
The output must be in a strict JSON
format: {"tree": "tree"}
TEXT: [TEXT]
RESPONSE: [RESPONSE]
```

The global representation serves as a map within which specific local details can be situated and clusters related elements together, thereby avoiding the presence of discrete details. Furthermore, it establishes a robust foundation for subsequent local path identification and detection.

## 3.3 Local Path Identification

We identify local details from the global representation constructed in Section 3.2. These local details are crucial for performing targeted detection of specific text segments. Specifically, we adopt a path-wise identification algorithm to achieve this goal, with each path considered as a local detail.

**Path Definition** A path is a sequence of connected nodes within the hierarchical structure, starting from the root node and traversing through intermediate nodes to reach a specific leaf node. Each path represents a unique contextual flow from the global theme to a specific detail within the text, thus enabling independent detection.

**Traversal Strategy** To identify local paths, an efficient traversal strategy that ensures comprehensive coverage is essential. We achieve this by employing a path-wise traversal strategy that efficiently explores the hierarchical structure. The strategy collects the nodes from the root to leaf nodes as the local details. This allows us to thoroughly explore each branch of the hierarchical tree structure, ensuring that all potential paths are identified.

The path-wise identification method covers all details in the global representation of the hierarchical tree structure, preventing missed detection and enabling fine-grained hallucination detection. After performing path identification on the global representation $\mathcal{G}$, we obtain $m$ local details $\mathcal{P} = \{p_1, p_2, \cdots, p_m\}$ to be detected.

## 3.4 Detection and Aggregation

We conduct detection on each path-wise detail and aggregate results with a logical operation to reach a final conclusion.

**Path-wise Detection** Path-wise detection begins by examining each identified local path within the hierarchical structure for potential hallucinations against reference documents. Similarly, we use pre-trained language models to achieve this goal. This process can be formulated as:

$$r_i = f_\theta(p_i, T_{dec}) \tag{2}$$

where $f_\theta(\cdot)$ is a language model parameterized by $\theta$, $p_i$ denotes the $i$-th pathwise local detail to be detected, $T_{dec}$ is detection prompt shown in Section 4.3, and $r_i$ is the response of the language model $f_\theta(\cdot)$.

After performing detection for $i$-th pathwise details, we get the $l_i$ from the response $r_i$, which indicates whether it contains a hallucination. We perform detection using Equation 2 on each pathwise local detail in $\{p_1, p_2, \cdots, p_m\}$ then obtain $m$ labels $\{l_1, l_2, \cdots, l_m\}$.

**Aggregation** After individual pathwise detection, the results are aggregated to form a comprehensive assessment. The aggregation takes as input a logical expression that performs AND operation over the variables in $m$ pathwise detail labels $\{l_1, l_2, \cdots, l_m\}$. This process can be defined as:

$$l = l_1 + l_2 + \cdots + l_m \quad (3)$$

where $+$ denotes the AND operation and $l$ denotes final hallucination detection conclusion.

# 4 Experiments

## 4.1 Datasets and Evaluation

To demonstrate the effectiveness of our proposed approach, we conduct experiments on four hallucination detection datasets: **HaluEval-Summary** in benchmark HaluEval (Li et al. 2023), **QAGS-CNNDM** and **FEVER** in TRUE benchmark (Honovich et al. 2022), and **SCIFACT** (Wadden et al. 2022). In our experiments, due to the resource limitations associated with using LLM, we sample a portion from each dataset, following previous methods (Wei et al. 2022). For the HaluEval-Summary dataset, we extract a sample at a ratio of one-tenth. In the QAGS-CNNDM dataset, we only remove samples that contain sensitive vocabulary. For the FEVER dataset, we select only those samples where the text to be detected exceeds 20 tokens. For SCIFACT, we exclude samples from the original dataset where the reference documents are absent. Detailed information of the datasets is presented in Table 1.

| **Dataset** | **Type** | **Sample** (pairs) | **Tokens** (avg) | **Sents** (avg) |
|---|---|---|---|---|
| HaluEval-S | News | 1,000 | 61.5 | 4.0 |
| QAGS-C | News | 233 | 48.9 | 3.0 |
| FEVER | News | 110 | 22.5 | 1.0 |
| SCIFACT | Science | 93 | 11.5 | 1.0 |

Table 1: Detailed information on the dataset used in our paper. HaluEval-S and QAGS-C are abbreviations HaluEval-Summary and QAGS-CNNDM, respectively. Sents (avg) and Tokens (avg) denote the average number of sentences and tokens, respectively.

We conduct a thorough evaluation of our proposed method across all selected datasets, using macro-average results as our metric. This approach allows us to assess the effectiveness of our proposed method uniformly by averaging the performance metrics, such as precision, recall, and F1 score, across different test cases and data types.

## 4.2 Baselines

To evaluate the performance of our proposed method, we compare it with several existing baseline methods. These baseline methods provide a benchmark for evaluating the relative performance of our proposed approach. We compare our method with 6 baselines, categorized into two groups.

**(1) End2End.** **Standard** and **Chain of Thought (CoT)** (Wei et al. 2022) methods allow the language model to handle the entire text at once, and we consider these two approaches as end-to-end methods. The CoT method involves augmenting the standard method by adding a step-by-step thought process. In our paper, we implement the CoT method by appending the sentence "Let's think step by step." at the end of the instruction.

**(2) Text2Local.** **SelfCheckGPT** (Manakul, Liusie, and Gales 2023) splits the text as sentences according to the grammatical rules, then aggregates the sentence detection results. The **FactScore** (Min et al. 2023) utilizes grammatical rules to segment the text into sentences, which are then further broken down into sub-sentences through the in-context learning capabilities of LLMs. **Factool** (Chern et al. 2023) is a comprehensive factuality detection and verification framework. Different from SelfCheckGPT and FActScore, FacTool directly extracts the sub-sentence from text using the in-context learning ability of LLMs. **REFCHECKER** (Hu et al. 2024) extracts triples from the text and then performs fine-grained hallucination detection. These methods extract local details from the original text and then aggregate them to produce a final result, hence we categorize them as Text2Local methods.

## 4.3 Implementation Details

In our paper, the LLMs used in global representation and detection include LLAMA3-8B-Instruct (Meta 2024), ChatGPT (GPT-3.5-Turbo-0613) (OpenAI 2022) and GPT-4 ( GPT-4-0613) (OpenAI 2024). To ensure reproducibility, for the parts involving the large model, we configure all models with the top_p parameter as 1.0 and temperature as 0.0. The prompts used to obtain local details in the baseline method are taken from the original paper, but are set to zero-shot. The specific detection prompts used in the experiments for the baseline methods and our method are as follows:

```
Document: [DOCUMENT]
Based on the above information, is it
true that [DETAILS]? True or False?
The answer is:
```

The baseline methods and our G2LDetect both adopt a zero-shot setting to counteract the potential randomness associated with demonstrations in a few-shot setting.

# 5 Results and Analysis

## 5.1 Overall Performances

**Results on Long Text** Table 2 presents a comprehensive performance comparison between our G2LDetect and existing techniques on HaluEval-Sumamry (Li et al. 2023) and QAGS-CNNDM (Honovich et al. 2022) datasets. These two datasets contain longer texts, which brings challenges for hallucination detection methods. Nevertheless, our G2LDetect still achieved the best results on ChatGPT, LLAMA3-7B, and GPT-4 across open and closed

| **LLMs** | **Methods** | **HaluEval-S** | | | **QAGS-C** | | |
|---|---|---|---|---|---|---|---|
| | | **F1** | **Precision** | **Recall** | **F1** | **Precision** | **Recall** |
| LLAMA3-8B-Instruct | *End2End Methods* | | | | | | |
| | Standard | 54.69 | 59.97 | 57.50 | 51.41 | 61.89 | 57.04 |
| | CoT | 54.90 | 63.45 | 58.80 | 53.46 | 73.76 | 60.68 |
| | *Text2Local Methods* | | | | | | |
| | SelfCheckGPT | 57.90 | 57.90 | 57.90 | 62.89 | 63.71 | 63.27 |
| | FactScore | 55.78 | 62.70 | 59.00 | 52.15 | 60.45 | 56.30 |
| | FacTool | 60.76 | 61.70 | 61.00 | 64.23 | 64.38 | 64.21 |
| | REFCHECKER | 55.59 | 58.02 | 57.00 | 63.80 | 64.68 | 64.29 |
| | **G2LDetect** | **64.34** | **64.82** | **64.00** | **67.24** | **70.33** | **68.72** |
| ChatGPT | *End2End Methods* | | | | | | |
| | Standard | 61.63 | 71.63 | 64.70 | 52.81 | 73.51 | 60.26 |
| | CoT | 62.82 | 69.59 | 65.00 | 49.99 | 57.18 | 54.77 |
| | *Text2Local Methods* | | | | | | |
| | SelfCheckGPT | 65.88 | 66.23 | 66.00 | 62.63 | 66.14 | 64.21 |
| | FactScore | 66.84 | 67.34 | 67.00 | 62.97 | 63.33 | 63.22 |
| | FacTool | 67.88 | 68.26 | 68.00 | 63.39 | 66.99 | 65.00 |
| | REFCHECKER | 67.16 | 70.05 | 68.00 | 64.00 | 64.46 | 64.08 |
| | **G2LDetect** | **70.97** | 71.08 | **71.00** | **68.75** | 72.15 | **69.31** |
| GPT-4 | *End2End Methods* | | | | | | |
| | Standard | 65.88 | 74.46 | 68.10 | 73.74 | 80.82 | 75.52 |
| | CoT | 66.77 | 75.40 | 68.90 | 72.94 | 79.25 | 74.63 |
| | *Text2Local Methods* | | | | | | |
| | SelfCheckGPT | 69.81 | 70.53 | 70.00 | 78.81 | 80.53 | 78.97 |
| | FactScore | 70.33 | 73.08 | 71.00 | 79.01 | 79.03 | 79.00 |
| | FacTool | 70.93 | 71.21 | 71.00 | 77.89 | 79.34 | 78.02 |
| | REFCHECKER | 69.00 | 72.98 | 70.00 | 72.35 | 72.64 | 72.33 |
| | **G2LDetect** | **72.98** | **73.08** | **73.00** | **84.44** | **88.81** | **84.38** |

Table 2: Evaluation results on datasets with long text. HaluEval-S and QAGS-C are abbreviations for HaluEval-Summary and QAGS-CNNDM, respectively.

LLMs Specifically, our method using ChatGPT achieved F1 scores of 70.97% and 68.75% on the HaluEval-Summary and QAGS-CNNDM datasets, respectively. After employing GPT-4, the F1 scores further improved to 72.98% and 84.44%. The CoT (Wei et al. 2022) is well-known for its effectiveness; however, its performance is quite limited on the HaluEval-Summary dataset. On the QAGS-CNNDM dataset, its performance even declines compared to the standard method. This indicates that although the CoT method can encourage deeper reasoning, it struggles with accuracy when dealing with long texts.

Local details extraction-based methods such as SelfCheckGPT (Manakul, Liusie, and Gales 2023), FactScore (Min et al. 2023), FacTool (Chern et al. 2023), and REFCHECKER (Hu et al. 2024) show significant improvements over standard methods, but there is still scope for improvement compared to our G2LDetect. Additionally, rule-based extraction methods such as SelfCheckGPT and FactScore are generally less effective than methods based on the in-context learning abilities of LLMs, such as FacTool. This indicates that rule-based extraction methods have inherent limitations. REFCHECKER provides finer-grained hallucination detection results, but the results are inferior to sentence or sub-sentence level methods on GPT-4.

**Results on Complex Text** We verify our methods on a dataset FEVER (Honovich et al. 2022) with multi-hop characteristics, the results are shown in Table 3. The results indicate that our G2LDetect still performs well on texts requiring multi-hop reasoning. For example, using ChatGPT and GPT-4, G2LDetect achieved F1 scores of 68.42% and 75.55%, respectively, which is an improvement of 3.77% and 5.44% over the Standard method. Compared to the leading methods based on local detail extraction, our method improved by 2.51% on ChatGPT and 2.99% on GPT-4, further

| LLMs | Methods | FEVER | | | SCIFACT | | |
|---|---|---|---|---|---|---|---|
| | | F1 | Precision | Recall | F1 | Precision | Recall |
| LLAMA3-8B-Instruct | *End2End Methods* | | | | | | |
| | Standard | 67.60 | 69.22 | 71.79 | 86.02 | 86.35 | 86.18 |
| | CoT | 68.50 | 70.28 | 70.79 | 83.11 | 83.12 | 83.11 |
| | *Text2Local Methods* | | | | | | |
| | SelfCheckGPT | 68.52 | 69.84 | 72.53 | 86.02 | 86.35 | 86.18 |
| | FactScore | 61.25 | 61.99 | 60.92 | 72.07 | 78.97 | 73.92 |
| | FacTool | 65.47 | 65.16 | 66.33 | 79.22 | 82.80 | 80.07 |
| | REFCHECKER | 62.18 | 62.46 | 65.47 | 82.92 | 63.35 | 62.92 |
| | **G2LDetect** | **69.27** | **70.54** | 69.08 | **86.16** | 86.20 | **86.94** |
| ChatGPT | *End2End Methods* | | | | | | |
| | Standard | 62.65 | 69.11 | 70.38 | 88.15 | 88.10 | 88.13 |
| | CoT | 59.79 | 64.67 | 66.10 | 77.16 | 79.63 | 77.85 |
| | *Text2Local Methods* | | | | | | |
| | SelfCheckGPT | 62.65 | 69.11 | 70.38 | 88.15 | 88.20 | 88.13 |
| | FactScore | 62.25 | 62.08 | 63.12 | 82.78 | 83.26 | 82.99 |
| | FacTool | 59.91 | 60.55 | 62.10 | 86.02 | 86.11 | 86.11 |
| | REFCHECKER | 65.91 | 66.63 | 69.14 | 75.08 | 75.54 | 75.07 |
| | **G2LDetect** | **68.42** | 69.06 | 68.00 | 88.12 | **89.46** | **88.47** |
| GPT-4 | *End2End Methods* | | | | | | |
| | Standard | 70.11 | 71.60 | 74.86 | 93.55 | 93.57 | 93.61 |
| | CoT | 71.13 | 72.88 | 76.29 | 92.47 | 92.46 | 92.50 |
| | *Text2Local Methods* | | | | | | |
| | SelfCheckGPT | 72.64 | 73.31 | 76.86 | 91.40 | 91.60 | 91.53 |
| | FactScore | 66.58 | 66.37 | 66.85 | 82.55 | 85.82 | 83.26 |
| | FacTool | 70.21 | 69.70 | 71.62 | 89.25 | 89.45 | 89.38 |
| | REFCHECKER | 72.76 | 72.76 | 72.76 | 80.13 | 85.71 | 81.25 |
| | **G2LDetect** | **75.55** | **75.65** | 72.10 | **94.62** | **95.00** | **94.79** |

Table 3: Results on FEVER and SCIFACT dataset.

validating the effectiveness of our approach.

**Results on Other Domain** SCIFACT (Wadden et al. 2022) is a dataset in the scientific domain, which poses unique challenges due to its reliance on factual accuracy and technical specificity. The results presented in Table 3 show that existing direct detail extraction methods such as SelfCheckGPT, FactScore, and REFCHECKER do not improve performance on ChatGPT compared to the standard method, our method achieves 1.36% improvement in Precision. The similar observations in LLAMA3-8B-Instruct and ChatGPT are mirrored in GPT-4.

## 5.2 Analysis

**Effect of Global Representation** To validate the effectiveness of global representations in capturing factual units, we conducted experiments using LLAMA3-8B-Instruct and ChatGPT (GPT-3.5-Turbo) on two longer datasets, HaluEval-Summary (Li et al. 2023) and QAGS-CNNDM (Honovich et al. 2022). Specifically, we replace the text to be detected in the existing methods, FacTool (Chern et al. 2023) and REFCHECKER (Hu et al. 2024), with the global representations of the text, while keeping everything else consistent with the original methods. The results obtained are shown in Table 4. In the results table, methods with the global subscript indicate that the text to be detected in the original methods is replaced with the global representation.

From the results, it can be seen that in both the FacTool and REFCHECKER methods, extracting factual units after obtaining the global representation of the text leads to an average improvement of 2.2% in F1 score. In FacTool, the factual units are sentences, while in REFCHECKER, they are triples. This approach enhances the final detection results in both cases. For example, the REFCHECKER method utilizing global representation with the LLAMA3-8B-Instruct model achieved a 3.87% improvement in F1 score on the HaluEval-Summary dataset. Similarly, the FacTool method

| Methods | HaluEval-S | QAGS-C |
|---|---|---|
| LLAMA3-8B-Instruct | | |
| FacTool | 60.76 | 64.23 |
| REFCHECKER | 55.59 | 63.80 |
| **G2LDetect** | **64.34** | **67.24** |
| FacTool$_{global}$ | 62.88 | 66.04 |
| REFCHECKER$_{global}$ | 59.46 | 65.72 |
| ChatGPT | | |
| FacTool | 67.88 | 63.39 |
| REFCHECKER | 67.16 | 64.00 |
| **G2LDetect** | **70.97** | **68.75** |
| FacTool$_{global}$ | 69.08 | 65.74 |
| REFCHECKER$_{global}$ | 68.82 | 65.91 |

Table 4: F1 performance of existing methods with global representation on QAGS-CNNDM and HaluEval-Summary datasets. FacTool$global$ and REFCHECKER$global$ indicate that the text to be checked in the original methods has been replaced with a global representation.

using ChatGPT with global representation resulted in a 2.35% improvement in F1 score on the QAGS-CNNDM dataset. The consistent results between open-source and closed-source large language models demonstrate the effectiveness of global representations.

**Effect of Quantity of Local Details** To investigate the impact of the quantity of local details on the final conclusion, we conduct a detailed analysis of the results of GPT-4 on the HaluEval-Summary dataset (Li et al. 2023). Specifically, we statistic the average number of local details extracted by SelfCheckGPT (Manakul, Liusie, and Gales 2023), FactScore (Min et al. 2023), FacTool (Chern et al. 2023), REFCHECKER (Hu et al. 2024)and our G2LDetect, respectively. The results are shown in Figure 3.

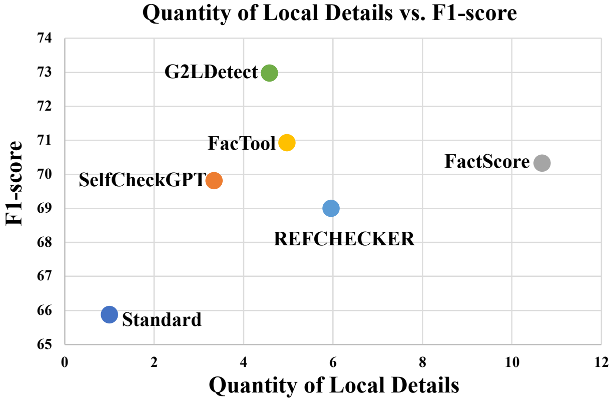


Figure 3: The quantity of local details extracted by different methods vs. F1-score.

The results show that the methods achieving better final f1-score results, specifically our G2LDetect and FacTool, contain the number of local details between 4 and 6. According to the statistics of the average number of sentences in the HaluEval-Summary dataset shown in Figure 3, the dataset contains an average of 4 sentences. Therefore, the method used by SelfCheckGPT, which directly splits sentences based on grammatical rules and treats them as local details, results in missed detection. The FactScore method further decomposes sentences into sub-sentence level local details, significantly increasing the number of local details. However, this approach only results in a slight improvement in performance, further indicating that extracting details based on grammatical rules is insufficient. REFCHECKER, although extracting triple-level local details and achieving more fine-grained detection, shows the least improvement in performance. This may be because triple-level local details lack sufficient contextual information.

**Human Evaluation** As shown in Figure 3, our method and FacTool have a similar average number of local details, yet achieve different F1 results. We conduct a further manual evaluation of their local details. Specifically, we invite two volunteers to annotate the 50 results of both methods and remove the invalid local details from them. Specifically, invalid details include: (1) Lack of Context Support (2) Irrelevant Information (3) Repetitive Information (4) Logical Break. Those invalid details cannot be meaningfully or accurately hallucination detected independently. Then we compare the average number of local details before and after human annotations. The results in Table 5 show that the average number of local details for Factool and our method decreased by 7% and 4%, respectively. The smaller decrease in our method indicates that the local details constructed by our path-wise methods are more effective.

| Methods | Numbers | | F1-score |
|---|---|---|---|
| | Before | After | |
| FacTool | 5.2 | 4.8 | 70.93 |
| G2LDetect | 4.8 | 4.6 | 72.98 |

Table 5: Comparison of the average number of local details before and after human annotations.

## 6 Conclusion

In this paper, we propose a global-to-local approach for hallucination detection. Our method first constructs a global representation of the original text and then adopts a path-wise identification algorithm to obtain local details for detection. The hierarchical tree structure global representation includes the main elements of the original text and clusters related information together, thus avoiding isolated details. The local details obtained by the path-wise identification algorithm cover all paths in the global representation, thereby avoiding missed detections. To demonstrate the effectiveness of our method, we conduct experiments on four datasets across two benchmarks. The results show that our proposed global-to-local approach outperforms existing methods, particularly in handling longer text.

## Acknowledgments

This work is supported by the Science Foundation of Donghai Laboratory (Grant No. DH-2022ZY0013), the National Natural Science Foundation of China (No. 62376245), the Fundamental Research Funds for the Central Universities (226-2024-00170), and MOE Engineering Research Center of Digital Library.

## References

Bang, Y.; Cahyawijaya, S.; Lee, N.; Dai, W.; Su, D.; Wilie, B.; Lovenia, H.; Ji, Z.; Yu, T.; Chung, W.; Do, Q. V.; Xu, Y.; and Fung, P. 2023. A Multitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity. 675–718. Nusa Dua, Bali.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 1877–1901. Curran Associates, Inc.

Chen, C.; Liu, K.; Chen, Z.; Gu, Y.; Wu, Y.; Tao, M.; Fu, Z.; and Ye, J. 2024. INSIDE: LLMs' Internal States Retain the Power of Hallucination Detection. arXiv:2402.03744.

Chern, I.-C.; Chern, S.; Chen, S.; Yuan, W.; Feng, K.; Zhou, C.; He, J.; Neubig, G.; and Liu, P. 2023. FacTool: Factuality Detection in Generative AI – A Tool Augmented Framework for Multi-Task and Multi-Domain Scenarios. arXiv:2307.13528.

Honovich, O.; Aharoni, R.; Herzig, J.; Taitelbaum, H.; Kukliansy, D.; Cohen, V.; Scialom, T.; Szpektor, I.; Hassidim, A.; and Matias, Y. 2022. TRUE: Re-evaluating Factual Consistency Evaluation. 161–175. Dublin, Ireland.

Hu, X.; Ru, D.; Qiu, L.; Guo, Q.; Zhang, T.; Xu, Y.; Luo, Y.; Liu, P.; Zhang, Y.; and Zhang, Z. 2024. RefChecker: Reference-based Fine-grained Hallucination Checker and Benchmark for Large Language Models. arXiv:2405.14486.

Ji, Z.; Lee, N.; Frieske, R.; Yu, T.; Su, D.; Xu, Y.; Ishii, E.; Bang, Y. J.; Madotto, A.; and Fung, P. 2023. Survey of Hallucination in Natural Language Generation. *ACM Computing Surveys*, 55(12): 1–38.

Li, J.; Cheng, X.; Zhao, X.; Nie, J.-Y.; and Wen, J.-R. 2023. HaluEval: A Large-Scale Hallucination Evaluation Benchmark for Large Language Models. 6449–6464. Singapore.

Manakul, P.; Liusie, A.; and Gales, M. 2023. SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models. 9004–9017. Singapore.

Meta. 2024. The Llama 3 Herd of Models. arXiv:2407.21783.

Min, S.; Krishna, K.; Lyu, X.; Lewis, M.; Yih, W.-t.; Koh, P.; Iyyer, M.; Zettlemoyer, L.; and Hajishirzi, H. 2023. FActScore: Fine-grained Atomic Evaluation of Factual Precision in Long Form Text Generation. 12076–12100. Singapore.

OpenAI. 2022. Introducing chatgpt.

OpenAI. 2024. GPT-4 Technical Report. arXiv:2303.08774.

Orru, G.; and Longo, L. 2018. The Evolution of Cognitive Load Theory and the Measurement of Its Intrinsic, Extraneous and Germane Loads: A Review. In Longo, L.; and Leva, M. C., eds., *Human Mental Workload: Models and Applications - Second International Symposium, H-WORKLOAD 2018, Amsterdam, The Netherlands, September 20-21, 2018, Revised Selected Papers*, volume 1012 of *Communications in Computer and Information Science*, 23–48. Springer.

Poliak, A. 2020. A survey on Recognizing Textual Entailment as an NLP Evaluation. 92–109. Online.

Qiu, Y.; Ziser, Y.; Korhonen, A.; Ponti, E.; and Cohen, S. 2023. Detecting and Mitigating Hallucinations in Multilingual Summarisation. 8914–8932. Singapore.

Su, W.; Wang, C.; Ai, Q.; Hu, Y.; Wu, Z.; Zhou, Y.; and Liu, Y. 2024. Unsupervised Real-Time Hallucination Detection based on the Internal States of Large Language Models. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Findings of the Association for Computational Linguistics ACL 2024*, 14379–14391. Bangkok, Thailand and virtual meeting: Association for Computational Linguistics.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; Bikel, D.; Blecher, L.; Ferrer, C. C.; Chen, M.; Cucurull, G.; Esiobu, D.; Fernandes, J.; Fu, J.; Fu, W.; Fuller, B.; Gao, C.; Goswami, V.; Goyal, N.; Hartshorn, A.; Hosseini, S.; Hou, R.; Inan, H.; Kardas, M.; Kerkez, V.; Khabsa, M.; Kloumann, I.; Korenev, A.; Koura, P. S.; Lachaux, M.-A.; Lavril, T.; Lee, J.; Liskovich, D.; Lu, Y.; Mao, Y.; Martinet, X.; Mihaylov, T.; Mishra, P.; Molybog, I.; Nie, Y.; Poulton, A.; Reizenstein, J.; Rungta, R.; Saladi, K.; Schelten, A.; Silva, R.; Smith, E. M.; Subramanian, R.; Tan, X. E.; Tang, B.; Taylor, R.; Williams, A.; Kuan, J. X.; Xu, P.; Yan, Z.; Zarov, I.; Zhang, Y.; Fan, A.; Kambadur, M.; Narang, S.; Rodriguez, A.; Stojnic, R.; Edunov, S.; and Scialom, T. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. arXiv:2307.09288.

Wadden, D.; Lo, K.; Kuehl, B.; Cohan, A.; Beltagy, I.; Wang, L. L.; and Hajishirzi, H. 2022. SciFact-Open: Towards open-domain scientific claim verification. 4719–4734. Abu Dhabi, United Arab Emirates.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; ichter, b.; Xia, F.; Chi, E.; Le, Q. V.; and Zhou, D. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems*, volume 35, 24824–24837. Curran Associates, Inc.

Zhang, Q.; Chen, S.; Xu, D.; Cao, Q.; Chen, X.; Cohn, T.; and Fang, M. 2023. A Survey for Efficient Open Domain Question Answering. 14447–14465. Toronto, Canada.