A Dynamic Approach to Detect Hallucination on LLM Summaries Generated Utilizing Term Overlap and Embedding Similarities

Alireza Nouri, Yura Posukhovskyi and Harsh Singhal

AI/ML CoE, WellsFargo & Co, 11625 N Community House Rd,
Charlote, 28277, North Carolina, USA.

*Corresponding author(s). E-mail(s): Pasha.Nouri@wellsfargo.com;
Contributing authors: Iurii.Posukhovskyi@wellsfargo.com;
Harsh.Singhal@wellsfargo.com;

Abstract

Large Language Models (LLMs) have demonstrated remarkable capabilities in text-generation tasks, including summarization. However, they generate hallucinated content—information that is not present in the original text. This work presents a novel, fully dynamic model for detecting hallucinations in summaries generated by LLMs. Unlike existing approaches that rely on user-defined parameters and thresholds, our model utilizes three core techniques: term overlap analysis, frequency-based term evaluation, and embedding similarity measurements between the original text and the generated summary. By detecting anomalies—textual elements introduced during summarization that are absent from the source—we effectively identify hallucinated content without requiring prior knowledge of the input document. Experimental results demonstrate that our method significantly surpasses baseline models, showing its robustness and adaptability in real-world applications where access to original documents may be limited for tuning and finding thresholds. Our approach represents a significant advancement in hallucination detection, enhancing the reliability of LLM-generated summaries across various domains.

Keywords: Large Language Models, Hallucination, Summarization, Text similarity

1 Introduction

This paper proposes a novel method to detect hallucinations in summaries generated by Large Language Models (LLMs). Hallucination detection is a vital study area [1–6] that focuses on enhancing the reliability of LLMs. This component makes LLMs more applicable to real-world and industry-scale applications [7–9]. Despite the advancements in text generation, hallucinations remain a notable challenge that limits the usage of LLMs in sensitive domains such as healthcare, finance, and legal applications [10–15].

1.1 Background and Motivation

Large Language Models (LLMs) brought more attention to natural language processing (NLP) by showing excellent text generation, summarization, machine translation, and question answering ability. Models such as GPT-4, BERT, T5, and BART transformers architecture, attention mechanisms, and big data to generate text with high quality and coherence [7, 16–23].

LLMs have been used in text summarization to automatically reduce large bodies of text into concise and informative summaries. This task is utilized in applications in news aggregation, legal document analysis, scientific paper summarization, and medical report generation [2, 3, 5, 6]. Despite their outstanding abilities, LLMs remain black-box models with limitations in content control [24, 25]. This limitation leads these models to factual inaccuracies and hallucinations in their outputs.

One of the major challenges in LLM-generated text is hallucination [26–28]. Hallucination appears when the model introduces information not present in the original text or generates factually incorrect content. Hallucinations in summarization tasks occur when:

1. The model fabricates facts that were never mentioned in the original document.
2. It alters key details, such as dates, names, or statistics.
3. It creates misleading causal relationships that misinterpret the original meaning.

These hallucinations inject serious risks in real-world applications [27–29]:

- Healthcare and Medical Summarization: Incorrect summaries of patient records, clinical trials, or medical research papers can lead to misdiagnosis or inappropriate treatment decisions.
- Legal and Financial Documents: Hallucinated content in contracts, court cases, or financial reports can result in legal disputes and economic losses.
- News and Journalism: Including non-existent or manipulated facts in news summaries can spread misinformation, influencing public opinion and policy decisions.

The unpredictability of hallucinations makes it difficult to rely on LLM-generated summaries. This challenge emphasizes the urgent need for hallucination detection tools that ensure factual consistency.

Detecting and mitigating hallucinations has become a critical research challenge due to the increasing dependence on LLMs for automated content generation [27, 30, 31]. Unlike minor grammatical or fluency errors, hallucinations undermine the credibility of AI-generated text, and it causes restrictions on its applicability in real-world applications [27–29]. Hallucination detection is essential for the following reasons:

- **Enhancing Trust and Reliability:** Providing LLM-generated content that aligns with the original text improves user trust in LLM-generated summarization techniques.
- **Industry Adoption:** Businesses and institutions require robust and reliable models that do not introduce false information, especially in high-risk environments like healthcare, finance, and law.
- **Regulatory Compliance:** Hallucination detection models will play a key role in ensuring compliance with data accuracy and misinformation laws as governments and organizations rule ethical guidelines for LLM-generated content.

1.2 Challenges and limitations in Hallucination Detection

Despite notable advances in large language models (LLMs), hallucination detection remains a complex and unresolved problem [26, 27, 31]. Existing hallucination detection methods face several challenges, including dependence on user-defined thresholds, high computational costs, the risk of introducing new hallucinations into their judgments, and difficulty adapting to different domains. Furthermore, current detection techniques, including rule-based, machine learning-based, embedding similarity, and fact-checking approaches, have fundamental limitations that limit their usefulness in real-world applications [1–4].

1.2.1 Challenges in Detecting Hallucinations:

1. **Dependence on Thresholds for Classification:** Many hallucination detection models rely on user-defined thresholds to classify text as faithful or hallucinated. These thresholds typically determine:
   (a) The minimum similarity score between the generated summary and the source text.
   (b) The maximum allowable variation from factual statements.

   However, setting appropriate thresholds is highly dataset-dependent and often requires extensive fine-tuning. This additional process makes these models difficult to generalize across various domains. A fixed threshold may either overestimate or underestimate hallucinated content that causes false positives or false negatives.

2. **High Computational Cost of Current Hallucination Detection Methods:** Many existing hallucination detection techniques depend on complex and advanced deep-learning models that require substantial computational resources [32–34]. These approaches include:
   (a) Deep learning-based classifiers that require large labeled datasets for training.
   (b) Embedding similarity models that perform costly pairwise comparisons between the source and generated text.
   (c) Fact-checking techniques that query external knowledge bases or run retrieval-based verification significantly increase processing time.

These expensive computational processes limit the scalability and feasibility of hallucination detection for real-world applications.

3. **Risk of Introducing New Hallucinations During Detection:** A significant challenge in hallucination detection is the potential to introduce new hallucinations while verifying the text generated by another model [32, 35, 36]. This occurs when:
   (a) LLM-assisted verification models attempt to ”correct” hallucinations, but they introduce false information during their judgment.
   (b) Fact-checking models rely on external databases that contain inaccurate or outdated information. This leads to the propagation of incorrect claims and facts.

   These methods introduce new layers of complexity and make it difficult to determine the factual content of generated summaries rather than eliminating hallucinations from a summary.

4. **Difficulty in Adapting Models to Different Domains and Datasets:** Hallucination detection models often struggle with domain adaptation. Text in different domains requires different evaluation strategies. Existing models frequently require retraining or fine-tuning when applied to new domains, and this additional process reduces their flexibility. A dynamic detection model that adapts to different datasets without manual adjustments is crucial for real-world applicability.

1.2.2 The Need for a Dynamic, Threshold-Free, and Cost-Efficient Hallucination Detection Model:

A novel approach needs to address the limitations of existing hallucination detection methods such as:

- Eliminates dependence on user-defined thresholds: This makes LLMs adaptable to various datasets and domains.
- Minimizes computational costs: It ensures the feasibility of these models for real-time applications.
- Reduces the risk of introducing new hallucinations: It guarantees the new model does not inject new layers of hallucinations during the verification process.
- Does not require external fact-checking databases: This makes LLM application in the scenarios when an external dataset is not available for the fact-checking process.
- Adapts dynamically: It adopts LLMs to different domains without requiring extensive retraining or fine-tuning.

To address these challenges, our proposed model introduces a cost-efficient, threshold-free, and adaptable approach for hallucination detection. By leveraging term overlap, frequency-based evaluation, and embedding similarities. Utilizing these components helps our method provide a robust hallucination detector that enhances LLM-generated summaries' reliability, scalability, and usability.

1.3 Contributions of This Work

Hallucination detection in Large Language Models (LLMs) remains a challenging problem due to the computational cost, reliance on pre-defined thresholds, and difficulty adapting to diverse textual domains. This work introduces a novel hallucination detection model that addresses these limitations by offering a cost-efficient, threshold-free, and adaptable solution. Our method leverages multiple similarity measures, dynamically adjusts to different text types, and introduces an optional neural network component for enhanced flexibility. Below, we demonstrate the key contributions of our proposed model:

1. Cost-Efficient architecture of the model: It requires minimal computational resources compared to other existing models and can be embedded into real-time applications without extreme additional processing.
2. Threshold-Free Dynamic Approach: It eliminates manual threshold tuning for classifying text. This mechanism allows the model to adapt automatically to different datasets without the need for any additional steps.
3. Compatibility with Various Corpora: Its design supports flexible similarity measures. This flexibility enables the architecture to enhance its performance as new similarity measurements emerge in NLP.
4. Hallucination Detection in Unseen Texts: This model works effectively on previously unseen data due to not using any fixed threshold. This ability reduces the need for labeled datasets and fine-tuning steps in the hallucination detector pipeline.
5. Optional Neural Network Component: This optional step allows the model to automate determining the weighting of similarity measures. This additional component optimizes hallucination detection performance for specific domains.

These contributions demonstrate that our model is a practical, scalable, and accurate approach to hallucination detection in LLM-generated summaries.

1.4 Problem Description:

We introduce the mathematical framework for detecting hallucinated content to define the problem of hallucination in LLM in automatic summarization tasks.

Let's consider T to represent the original text where $t_i, ..., t_n$ are sentences in T and S to represent the summary generated by an LLM where $s_j, ..., s_m$ are sentences in the Summary S.

We define a hallucination as any information s_k in S that does not have a direct or semantically consistent representation t_l in T. We define hallucination as follows:

$$\begin{aligned} &T = \{t_1, t_2, ..., t_n\}, \\ &S = \{s_1, s_2, ..., s_m\} \\ &Where: \quad s_j \notin T \end{aligned} \tag{1}$$

This indicates that s_j does not appear in the original text T, meaning the presence of extrinsic hallucination. However, hallucinations are not just lexical mismatches; they may also appear as semantic distortions where the LLM rephrases existing content incorrectly or misinterprets relationships between entities.

In the sentence level, hallucination appears when a sentence in S contains information that is not inferred from or opposes the original text T. Define the semantic similarity function between a sentence $s_j \in S$ and the original text T as:

$$Similarity(s_j, T) = \max_{t_i \in T} Similarity(s_j, t_i) \tag{2}$$

Where, Similarity can be computed by using different term-based or context-based similarity measures such as cosine similarity, Jaccard similarity, or embedding-based methods. A sentence s_j is classified as hallucinated if:

$$Sim\,(s_j, T) < \delta \tag{3}$$

Where δ is a predefined threshold that our model eliminates dynamically.

The document-level hallucination representation quantifies the overall hallucination rate in a generated summary by comparing all its sentences against the original text. Instead of detecting hallucination on a word level, which is too restrictive and might ignore contextual consistency, this approach provides a global measure of how much hallucinated content exists in the summary.

We define the hallucination ratio $H(S, T)$ as:

$$H(S,T) = \frac{|\{s_j \in S \mid Sim(s_j, T) < \delta\}|}{|S|} \tag{4}$$

Where it counts the number of hallucinated sentences over the total number of sentences in the generated summary S. A higher $H(S, T)$ indicates a more hallucinated summary.

2 Related Works

2.1 Hallucination in Large Language Models

Hallucination in large language models (LLMs) refers to the generation of text that is either factually incorrect or not based on the original input [3–6]. This is a critical challenge for summarization, machine translation, and question-answering tasks, where factual consistency is essential [10–15]. In this section, we review existing studies on hallucination in LLMs.

1. **Definition and Types of Hallucination (Intrinsic vs. Extrinsic):**
   Literature categorized hallucinations in LLMs into intrinsic and extrinsic types based on their relationship with the input text [37–39].

   - Intrinsic Hallucination can occur when the generated text falsifies information in the input. For example, an LLM might change numerical values, misrepresent entities, or introduce incorrect relationships between different concepts and entities in summarization.

- Extrinsic Hallucination can occur when the generated output has some facts that are completely absent from the source text. This type of hallucination is particularly problematic in summarization tasks, where the model introduces facts not mentioned in the original document.

Both types of hallucination undermine the reliability of LLM, especially in domains such as healthcare, finance, and legal applications, where factual precision is critical.

2. **Challenges in Detecting Hallucinated Content in Text Generation:** Detecting hallucinations in LLM-generated content is naturally difficult due to some critical challenges:

- Semantic Paraphrasing: Hallucinated content can be semantically similar to original text information, making it difficult to differentiate using traditional overlap-based evaluation metrics such as ROUGE [40] or BLEU [41].
- Domain-Specific Constraints: Some hallucinations may be mild or contextually acceptable in creative writing, while they can be highly misleading in finance or medical applications [28].
- Model Bias and Training Data Artifacts: LLMs often inject biases from their training data, resulting in routine hallucinations in specific domains. Without external validation techniques, these biases remain concealed [42].

These challenges emphasize the need for robust and adaptive hallucination detection models that do not rely on pre-defined thresholds or manually designed rules.

3. **Overview of Studies that Analyze Hallucinations in LLMs:** Researchers are focusing on techniques to analyze and quantify hallucination in LLM-generated text. Early research focused on statistical and lexical overlap methods, such as ROUGE, to assess the faithfulness of the content [43, 44]. However, using just these metrics was verified insufficient to capture semantic and factual inconsistencies.

- Embedding-Based Similarity Approaches: Studies have explored using embedding-based techniques, such as BERTScore [45] and cosine similarity, to measure alignment between generated text and source content. Although these methods improve lexical overlap metrics, they still struggle with factual errors.
- Fact-Checking with External Knowledge Bases: Recent work has proposed integrating external databases (e.g., Wikipedia, knowledge graphs) to verify factual consistency [5, 46]. However, these approaches often require extensive labeled data and domain-specific knowledge, causing limits on generalizability.
- Model-Based Detection Strategies: Some researchers have trained classifiers to differentiate between hallucinated and faithful text. These models typically require supervised datasets and extensive parameter tuning, making them impractical for real-world deployment.

Despite these efforts, hallucination detection is still an unsolved problem, making LLMs less trustworthy. Our proposed approach addresses these challenges by utilizing term overlap, frequency analysis, and embedding similarities in a dynamic, threshold-free framework that enhances usability across different domains and applications.

2.2 Existing Approaches to Hallucination Detection:

Hallucination detection in text generated by an LLM has recently become one of the main focuses of researchers [1–6]. Various methods are suggested to identify and mitigate hallucinations. These approaches can be categorized into rule-based algorithms, machine learning-based models, and embedding-based similarity approaches. However, these methods have limitations, especially in their reliance on pre-defined thresholds, labeled data, or external knowledge sources [3, 5].

1. **Machine Learning-Based Models (Supervised Classifiers, Adversarial Detection):** Machine learning-based approaches have been widely used to detect hallucinations by training classifiers on labeled datasets [6, 47]. These models are trained using a mixture of real and hallucinated text. This training allows them to learn patterns associated with factual inconsistencies causing hallucinations in text generated by an LLM.

   - Supervised Learning Approaches: Many studies utilize neural network-based classifiers, such as transformers and LSTMs, to differentiate between factual and hallucinated content [27, 48]. Training these models requires labeled data in which humans annotated each text as faithful or hallucinated. Models like T5, GPT-3, and BART have been fine-tuned to improve their ability to detect hallucinations.
   - Adversarial detection: Some researchers proposed adversarial networks, where a secondary model (a discriminator) is trained to differentiate between factual and hallucinated text generated by an LLM [47, 49, 50]. This technique is highly effective in improving the robustness of the model but requires significant computational resources.
   - LLM-as-a-judge: Recent studies have leveraged an LLM to evaluate the performance of another LLM. This technique uses the language understanding of an LLM to detect hallucinations on text [51–53]. Learning this technique improves hallucination detection without relying on explicit labels. A few-shot approach will enhance the performance of the judge. However, the judge can still inject hallucination into its judgment.

   Although these machine-learning approaches have shown optimistic results, their dependence on labeled training data and the need for extensive fine-tuning question their abilities in real-world applications.

2. **Embedding-Based and Similarity-Based Approaches:** Embedding-based similarity techniques are another approach to detecting hallucination by measuring the similarities between generated summaries and their original texts [54, 55]. These methods utilize vector representations of text to explore key semantic elements from text.

- BERTScore: This metric computes the token-wise embedding similarity between the generated summary and the original document utilizing pre-trained transformer models like BERT [45]. Higher scores imply greater similarity whereas lower scores indicate less similarity between two texts.
- MoverScore: This technique extends embedding-based similarity by incorporating word importance and similarity between generated and original texts [56]. It has shown improvements over traditional metrics like ROUGE but is still sensitive to paraphrased hallucinations.
- Knowledge Graph-Based: This method incorporates knowledge graphs to verify whether entities and facts in the generated text align with external knowledge bases like Wikidata. However, this approach is highly domain-specific and requires access to structured external databases [57–59].

While embedding-based approaches enhance the ability to detect common hallucinations in text, they still fail to address most hallucinations, which are essential in some application domains like healthcare and finance. Additionally, these methods often require threshold tuning to differentiate between factual and non-factual content, making them unsuitable for most real-world applications.

3. **Limitations of Existing Models (Reliance on Thresholds, Need for Labeled Data):** Despite recent works on hallucination detection, existing models present limitations:

- Dependence on Labeled Datasets: Many supervised learning-based approaches require high-quality labeled datasets, which are expensive and time-consuming to collect. Also, labeled data may not be generalized well across different domains and may need experts to annotate.
- Threshold Sensitivity: Most current methods, particularly embedding-based approaches, depend on user-defined similarity thresholds to classify text as hallucinated or factual. These thresholds are often calculated upon a dataset and may not be adaptable for new datasets or real-world scenarios.
- Lack of Generalizability: Rule-based and knowledge graph-based methods are highly domain-specific and will fail when these approaches are applied to different applications outside their training corpus.
- Computational Costs: Adversarial training methods often require significant computation, making them unusable for large-scale applications.

To address these challenges, our proposed model introduces a fully dynamic, threshold-free approach that does not require labeled training data or user-defined parameters. By leveraging term overlap, frequency-based evaluation, and embedding similarities, our approach provides a more adaptable, scalable, and cost-friendly solution for hallucination detection in LLM-generated summaries.

2.3 Evaluating Summarization Faithfulness:

The faithfulness of summaries generated by an LLM is crucial for preserving the reliability and usability of summarization models. A loyal summary of the original text should accurately reflect the essential facts from the original text without presenting misinformation or hallucinated content [60, 61]. Researchers have proposed various evaluation methods to measure summaries' faithfulness, such as traditional lexical-based metrics, semantic-based approaches, and fact-checking techniques.

1. **Traditional Metrics (ROUGE, BLEU) and Their Limitations in Hallucination Detection:** Traditional summarization evaluation methods mainly rely on the lexical overlap between the generated summary and the original text. The most widely employed metrics include:

   - ROUGE (Recall-Oriented Understudy for Gisting Evaluation): ROUGE [40] calculates the overlap of n-grams, word sequences, and word pairs between the generated and original text. Variants such as ROUGE-N (n-gram overlap), ROUGE-L (longest common subsequence), and ROUGE-S (skip-bigram matching) are commonly used to evaluate summarization quality.
   - BLEU (Bilingual Evaluation Understudy): This metric [41] was originally developed for machine translation. BLEU evaluates the precision of n-gram matches between the generated summary and original text. It is useful for measuring fluency but does not account for factual accuracy.

   Limitations of ROUGE and BLEU in Hallucination Detection:

   - Both ROUGE and BLEU rely on exact word or phrase matching. This approach ignores semantic meaning and factual correctness in text. A summary may receive a high score despite containing incorrect information if it retains similar phrasing.
   - These metrics fail to capture meaning-preserving paraphrases or slight textual modifications that are common on text generated by an LLM. This issue limits their effectiveness in detecting factual inconsistencies.
   - Since lexical overlap solely cannot be used to capture factual content, these metrics cannot distinguish between summaries that faithfully represent the original text and those that contain hallucinated information.

   Due to these limitations, some works focus on utilizing semantic similarity-based and fact-checking approaches to improve the performance of hallucination detection models.

2. **Semantic Similarity-Based Approaches (BERTScore, MoverScore):** To address the limitations of previous metrics, researchers have developed embedding-based similarity techniques that evaluate the semantic consistency between a generated summary and the original text.

   - BERTScore: This method computes the similarity between each token in the generated summary and the most relevant token in the original text. This technique uses contextual embeddings from transformer-based models like BERT. BERTScore considers meaning and contextual similarity instead of relying on exact word matches. This approach makes it more effective for paraphrased content.

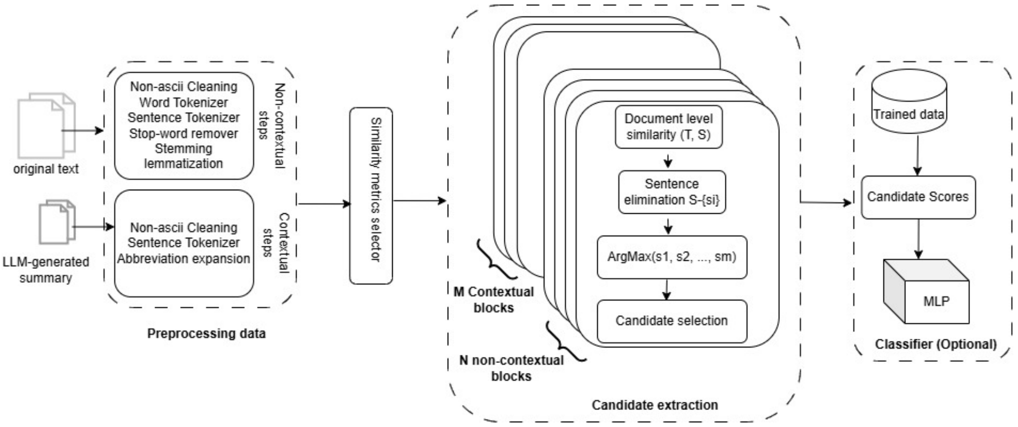

Fig. 1 The architectural Overview of the Proposed Hallucination Detection Model consists of three main components: (1) the Preprocessing Component, (2) the Candidate Extraction Component, and (3) the Classifier Component, which is optional.

- MoverScore: An extension of BERTScore, MoverScore uses optimal transport theory to calculate the minimum effort required to match words in the summary with words in the original text. MoverScore improves upon traditional embedding-based similarity metrics by considering word importance and overall semantic distance.

Limitations of Semantic Similarity-Based Approaches:

- While these approaches measure semantic similarity, they do not explicitly detect whether the information is factually correct or introduced erroneously. This failure appears when a hallucinated statement may still be semantically similar to the original text but factually incorrect.
- Considering that embeddings capture meaning better than lexical metrics, subtle factual alterations (e.g., changing a date or numerical value) may not significantly affect similarity scores. This leads to situations when hallucinations are left undetected.
- These methods often rely on user-defined similarity thresholds to classify summaries as faithful or hallucinated. This threshold is usually calculated based on a dataset, making them dataset-dependent.

Despite these challenges, semantic similarity-based metrics provide a more robust alternative to ROUGE and BLEU. However, researchers have explored fact-checking approaches incorporating external knowledge to enhance hallucination detection models.

3 Methodology

In this section, we explain the method of our proposed model for detecting hallucinations in LLM-generated summaries. Our approach is developed to address the main limitations in existing hallucination detection methods without needing any predefined threshold, capable of adopting any domain-based corpus, and without requiring an intense process. The architecture of our proposed model is illustrated in Figure 1.

The model consists of four main components to provide accurate and scalable hallucination detection:

1. **Preprocessing Component:** This module cleans, tokenizes, and normalizes both the original text and the generated summary, ensuring consistency in format and structure before similarity analysis. This process removes irrelevant symbols, stopwords, and redundant text that may interfere with accurate similarity measurements.
2. **Similarity Measurement Component:** This component calculates multiple semantic and lexical similarity metrics between the original text and the generated summary. It includes embedding-based similarity (e.g., BERTScore, cosine similarity), frequency-based overlap (e.g., TF-IDF), and term overlap (e.g., n-gram, Jaccard index) to deliver a comprehensive evaluation of potential hallucinations in text.
3. **Hallucination Candidate Harvester:** This module systematically identifies candidate sentences within the summary that factually conflict with the original text based on the similarity measures. It dynamically candidate sentences that reduce the similarity between the summary and the original text without requiring any similarity threshold.
4. **Neural Network-Based Decision Module:** The final step utilizes a neural network to dynamically weigh different similarity measures and classify sentences as hallucinated or faithful. The network learns to optimize detection performance based on training data, providing a flexible and adaptive decision mechanism that improves accuracy across different domains. This component is an optional step to enhance the performance of our algorithm by adopting it to any domain-based corpora.

By integrating these components, our methodology ensures robust threshold-free hallucination detection capable of handling various texts in different domains and reducing the risk of misclassification. The following sections provide a detailed breakdown of each component.

3.1 Preprocessing data

The preprocessing component ensures that both the original text and the LLM-generated summary are cleaned and standardized before further processing. This step removes inconsistencies and irrelevant elements that could interfere with text analysis to enhance the accuracy of similarity measurements. Key preprocessing operations include: 1) Text Cleaning: Eliminating unnecessary elements such as URLs, emojis, special characters, and non-ASCII symbols to maintain textual consistency. 2) Abbreviation Expansion: Replacing abbreviations and symbols with their full forms to ensure a uniform representation of meaning. 3) Handling Encodings: Converting text into a consistent encoding format (e.g., UTF-8) to avoid issues with character mismatches.

Tokenization is a critical step that segments the text into words and sentences, which enables accurate alignment between the generated summary and the original text. Our proposed model utilizes both word-level and sentence-level tokenization to optimize the effectiveness of similarity measurements in our proposed model.

- Word-Level Tokenization: In this technique, we break down text into individual words to guarantee that lexical similarity metrics can operate effectively.
- Sentence-Level Tokenization and Mapping: Our model dynamically maps summary sentences to relevant segments of the original text instead of treating each sentence independently.

Unlike traditional sentence segmentation, our approach does not strictly follow punctuation-based sentence splitting. Instead, it uses contextual similarity to group multiple sentences from the original text into logical chunks where they contribute to the meaning of a single summary sentence generated from them. This technique allows for a more accurate reference mapping between the original text's summary and source.

To achieve this, we utilize contextual similarity metrics (e.g., BERT-based embeddings) to determine which sentences in the original text are semantically linked to a given summary sentence. If multiple sentences in the original document contribute to the meaning of a single summary sentence, they are merged into a unified reference chunk.

This adaptive mapping enhances: 1) Sentence mapping accuracy to ensure that hallucinated content can be properly identified. 2) Similarity metric performance to reduce fragmentation and capture contextual dependencies more effectively. 3) Robustness of hallucination detection, particularly for complex summaries that condense multiple ideas from different parts of the source text. By utilizing context-aware sentence tokenization, our method delivers a more intelligent and flexible framework for hallucination detection. This approach ensures that every summary sentence is correctly mapped to its most relevant source context.

Let's consider $T = \{t_1, t_2, ..., t_n\}$ to be the set of sentences in the original text, and $S = \{s_1, s_2, ..., s_m\}$ to be the set of sentences in the generated summary. Hence, $Sim(t_i, s_j)$ represents the semantic similarity score between a sentence t_i from the original text and a summary sentence s_j where some contextual similarity metrics can compute it.

$$C_j = \{t_i \in T \mid Sim(t_i, s_j) \geq \delta\} \tag{5}$$

Where C_j is the set of original text sentences that contribute to the meaning of the summary sentence s_j and δ is a dynamic similarity threshold that ensures only semantically relevant sentences are merged.

$$T_j^{merged} = \bigcup_{t_i \in C_j} t_i \tag{6}$$

Where T_j^{merged} represents the combined text fragment aligned with s_j and the union operator $\bigcup$ merges all selected sentences into a single text unit for further analysis. We now define the final sentence tokenization function $f(T)$, which converts the original text into tokenized segments mapped to the summary:

$$f(T) = \{T_1^{merged}, T_2^{merged}, ..., T_m^{merged}\} \tag{7}$$

Where each tokenized segment T_j^{merged} contains one or more original sentences that contribute to the summary sentence s_j.

In addition to the general preprocessing applied to both the original text and the summary, an additional preprocessing step is specifically designed for term-based similarity metrics. This step reduces noise and standardizes word representations to ensure more precise lexical similarity calculations. Key enhancements in this preprocessing step include stopword removal, lemmatization, and stemming. This extra preprocessing step is only applied to term-based similarity metrics and is not used for contextual similarity methods. Since contextual embeddings preserve semantic meaning, additional text normalization could negatively impact performance.

3.2 Similarity-Based Hallucination Candidate Extraction:

In this step, our model utilizes multiple similarity measures to systematically determine sentences in the LLM-generated summary withare most likely to contain hallucinations or misleading facts. By using a variety of similarity metrics, our approach enhances the robustness and efficiency of the hallucination detector. This approach provides comprehensive coverage of different textual aspects.

We present N-independent similarity-based evaluation blocks, where each block uses a different similarity metric to evaluate the alignment between the summary sentence and the original text. These N blocks compute N different hallucination probabilities for each sentence in the summary. These probabilities offer multiple perspectives on potential inconsistencies.

All blocks in this phase have the same architecture, and the only difference between them is the similarity metrics used as their core evaluation component. By utilizing a range of lexical, statistical, and embedding-based similarity measures (TF-IDF, Ngram-based similarities, contextual and embedding-based similarities), the model tries to detect hallucinations in different forms (e.g., fabricated facts, semantic distortions, or out-of-context generalizations) effectively.

The output of these N similarity-based blocks is then collected and sent to a neural network as the final decision-making stage to classify sentences as factual or hallucinated. This modular design allows the model to adapt dynamically to different text types while minimizing false positives and false negatives in hallucination detection.

3.2.1 Step1: Document-Level Similarity Computation:

The first stage of our methodology is to calculate the document-level similarity between the LLM-generated summary and the original text. This similarity score provides a global measure of alignment between the two texts.

This score serves as an inverse representation of their semantic distance. A higher similarity score (shorter distance) indicates that the summary closely aligns with the source document. On the other hand, a lower similarity score (longer distance) suggests potential hallucinations, omissions, or distortions in the summary. Each similarity-based hallucination candidate extraction block uses the same similarity metric for this initial document-level assessment to provide consistency across the evaluation process. The computed similarity score for each sentence is forwarded to the next stage to determine whether specific changes in the summary increase or decrease the semantic distance from the original text.

3.2.2 Step 2: Sentence Tokenization for Fine-Grained Analysis:

Each similarity-based hallucination detection block contains a sentence tokenization module to segment the summary into individual sentences. This step allows a more fine evaluation, allowing the model to evaluate whether a specific part of the summary diverges from the source document. Given a summary S, we tokenize it into a set of sentences: $S = s_1, s_2, ..., s_n$ where: n represents the number of sentences in the summary, and we have $s_i \neq s_j$ to ensure that each tokenized sentence remains distinct and unique.

This sentence-level segmentation is essential for identifying hallucinations. This step allows the model to track sentence-wise deviations rather than rely solely on a holistic comparison of documents. In the next stage, multiple similarity-based evaluations will be applied over these tokenized sentences to extract potential hallucination candidates for further classification.

3.2.3 Step 3: Sentence Elimination for Hallucination Identification:

In this step, our model uses a sentence elimination technique to evaluate the contribution of each summary sentence to the overall document similarity between the generated summary and the original text. This method evaluates how removing each individual sentence affects the semantic alignment between the two texts. It provides an awareness of whether a sentence is factual or can be a hallucination.

The model computes the document similarity between the original text and the summary with sentence s_i removed for each sentence $s_i \in S$ in the summary: $Sim(T, S - \{s_i\})$ where $S - \{s_i\}$ denotes the summary after excluding sentence i. This similarity is then compared to the similarity score when s_i was still part of the summary:

$$\Delta Sim_i = Sim\,(T, S) - Sim(T, S - \{s_i\}) \tag{8}$$

The key hypothesis underlying our proposed model is:

- If s_i is factual, its removal should cause a decrease in similarity because it contributes to the faithfulness of the summary.
- If s_j is hallucinated, its removal should cause an increase in similarity, as hallucinated content introduces noise that disrupts the alignment between the summary and the original text.

Hence, the decision rule of our hallucination detector is: s_i is likely factual if $\Delta Sim_i < 0$, and s_j is likely hallucinated if $\Delta Sim_j > 0$. The coherence between the summary and its original text is reduced where hallucinations act as noise. The model isolates hallucinated content to identify sentences that negatively impact semantic consistency by removing and re-evaluating each sentence. This technique provides a dynamic and threshold-free approach to hallucination detection that enhances robustness across different types of summaries and domains.

At the end of this process, the model generates N different ΔSim_i^N score for each sentence in the summary S, corresponding to the N different similarity measures blocks used in the previous stages. These scores quantify the impact of each sentence on the overall similarity between the summary and the original text. An *ArgMax* function will be applied over all sentences to find the highest value of ΔSim, representing the sentence with the highest probability of being a hallucination.

$$Candidate_{1,...,N} = argMax_{1,..,N}(\{\Delta Sim_1, \Delta Sim_2, ..., \Delta Sim_M\} \tag{9}$$

In the next step, $Candidate_{1,...,N}$ is evaluated to determine its role in the summary Whether it is classified as a part of text or a hallucination. This classification helps the model distinguish factual content from hallucinated information to improve the reliability of the summaries generated by LLMs. The hallucination similarities extracted for the candidate are then passed to the final decision-making step where a neural network combines these scores to make the final classification decision.

3.3 Final Decision-Making and Classification:

Our proposed model incorporates the N different ΔSim_i^N values, where each ΔSim_i^n represents the degree of a sentence that contributes to the similarity between the summary and the original text. Specifically, ΔSim_i captures the maximum probability of a sentence being hallucinated. To maximize the efficiency of this approach, the model leverages multiple similarity-based metrics.

Each score in this step corresponds to a different similarity metric, each emphasizing a unique linguistic characteristic, such as Semantic similarity (e.g., cosine similarity of embeddings), Contextual relevance (e.g., BERTScore, SBERT), Lexical overlap (e.g., ROUGE, Jaccard similarity), and Term frequency-based comparisons (e.g., TF-IDF weighting).

The model provides a comprehensive and multi-perspective evaluation of hallucinations in LLM-generated summaries by incorporating diverse similarity measures.

The classification mechanism offers flexibility in determining hallucinations based on different decision rules:

1. **Agreement-Based Classification:** A sentence is classified as hallucinated only if all similarity metrics agree that its removal increases document similarity.
2. **Ensemble Voting Classification:** A sentence is flagged as hallucinated if most similarity metrics indicate that its removal improves summary alignment.

This modular approach allows the model to adapt to different use cases and accuracy requirements. This classification module can be customized for various applications in other domains.

For a more adaptive and domain-specific hallucination detection, we introduce a neural network classifier that learns the optimal decision boundary:

$$h_i = f_\theta(\Delta Sim_i^1, \Delta Sim_i^2, ..., \Delta Sim_i^N) \tag{10}$$

Where, f_θ represents a neural network parameterized by θ and $h_i \in [0, 1]$ is the predicted probability of s_i is being hallucinated. If $h_i \geq \tau$ (a learned threshold), then s_i is classified as hallucinated, otherwise, it is classified as factual content.

Our model includes an optional neural network component for domain-specific fine-tuning that can be trained when a sufficient number of labeled samples are available. This network optimizes the classification process by learning the optimal weighting of different similarity measures for a specific domain and enhancing classification efficiency by aligning hallucination detection with domain-specific needs (e.g., medical, legal, and financial texts).

The model maintains flexibility, scalability, and efficiency by incorporating an adaptive neural network option while providing robust hallucination detection to different real-world applications.

4 Conclusion

Hallucination detection in LLM-generated summaries is a crucial challenge that affects the reliability, accuracy, and real-world applicability of Generative AI-based summarization models. We proposed a novel, threshold-free, and computationally efficient hallucination detection model that utilizes multiple similarity-based evaluations to identify misleading or falsified content in summaries.

Our approach incorporates document-level and sentence-level similarity evaluation metrics to dynamically measure how the removal of each sentence impacts overall text alignment. Employing a diverse set of N similarity metrics, such as semantic, contextual, and lexical-based measures, helps the model evaluate hallucinations from multiple perspectives. Our method's flexibility allows it to use either a rule-based ensemble approach (full agreement or majority voting) or an optional neural network classifier to enhance hallucination detection based on domain-specific data.

The key contributions of our model include:

1. A fully dynamic, threshold-free hallucination detection mechanism, eliminating user-defined thresholds.
2. A multi-similarity framework, providing robust and accurate hallucination classification.
3. Scalability across different domains, making it adaptable to various text types such as legal, medical, financial, etc.
4. Computational efficiency, enabling real-time or large-scale hallucination detection without using high-cost models.

We have demonstrated that hallucinated content damages the coherence between a summary and its original text. By detecting and removing this irrelevant content, the reliability of AI-generated summaries can be significantly improved. This research presents a practical and scalable solution to hallucination detection, making it possible for AI-generated summarization models in critical applications and domains.

Algorithm 1 Hallucination Detection in LLM-Generated Summaries

```
Require: $T$: Original text (set of sentences)
        $S$: Generated summary (set of sentences)
        $N$: Number of similarity metrics
        $Sim^k(T, S)$: Similarity function for metric $k$
        $\delta$: Dynamic threshold for classification
        $f_\theta$: Optional neural network classifier
Ensure: $Y$: Binary hallucination labels for each sentence in $S$
Step 1: Preprocessing
$T, S \leftarrow$ ConPreprocess($T$), ConPreprocess($S$)        ▷ Contextual preprocessing
$T, S \leftarrow$ NConPreprocess($T$), NConPreprocess($S$),    ▷ Non-contextual preprocessing
Step 2: Compute Initial Document Similarity
for $k = 1$ to $N$ do
    $Sim^k_{baseline} \leftarrow Sim^k(T, S)$        ▷ Compute baseline document similarity
end for
Step 3: Sentence Elimination and Similarity Computation
for each sentence $s_i \in S$ do
    for $k = 1$ to $N$ do
        $Sim^k_{new} \leftarrow Sim^k(T, S - \{s_i\})$
        $\Delta Sim^k_i \leftarrow Sim^k_{baseline} - Sim^k_{new}$
    end for
    Store $\Delta Sim^k_i$ values for sentence $s_i$
end for
Step 4: Hallucination Classification
for each sentence $s_i \in S$ do
    $vote \leftarrow \sum_{k=1}^{N} \nVdash(\Delta Sim^k_i > 0)$
    if $vote \geq \frac{N}{2}$ then                  ▷ Majority voting
        $y_i \leftarrow 1$                            ▷ Sentence is hallucinated
    else
        $y_i \leftarrow 0$                            ▷ Sentence is factual
    end if
end for
Step 5: Optional Neural Network Classification
if using neural network then
    for each sentence $s_i \in S$ do
        $h_i \leftarrow f_\theta(\Delta Sim^1_i, \Delta Sim^2_i, ..., \Delta Sim^N_i)$
        if $h_i \geq \tau$ then
            $y_i \leftarrow 1$                        ▷ Sentence classified as hallucinated
        else
            $y_i \leftarrow 0$                        ▷ Sentence classified as factual
        end if
    end for
end if
Step 6: Output Hallucination Labels
return $Y = \{y_1, y_2, ..., y_m\}$                  ▷ Final classification results
```

References

[1] Chen, C., Liu, K., Chen, Z., Gu, Y., Wu, Y., Tao, M., Fu, Z., Ye, J.: Inside: Llms' internal states retain the power of hallucination detection. arXiv preprint arXiv:2402.03744 (2024)

[2] Chen, Y., Fu, Q., Yuan, Y., Wen, Z., Fan, G., Liu, D., Zhang, D., Li, Z., Xiao, Y.: Hallucination detection: Robustly discerning reliable answers in large language models. In: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, pp. 245–255 (2023)

[3] Valentin, S., Fu, J., Detommaso, G., Xu, S., Zappella, G., Wang, B.: Cost-effective hallucination detection for llms. arXiv preprint arXiv:2407.21424 (2024)

[4] Yehuda, Y., Malkiel, I., Barkan, O., Weill, J., Ronen, R., Koenigstein, N.: Interrogatellm: Zero-resource hallucination detection in llm-generated answers. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 9333–9347 (2024)

[5] Sriramanan, G., Bharti, S., Sadasivan, V.S., Saha, S., Kattakinda, P., Feizi, S.: Llm-check: Investigating detection of hallucinations in large language models. Advances in Neural Information Processing Systems **37**, 34188–34216 (2025)

[6] Chakraborty, N., Ornik, M., Driggs-Campbell, K.: Hallucination detection in foundation models for decision-making: A flexible definition and review of the state of the art. ACM Computing Surveys (2025)

[7] Liu, Y., Yao, Y., Ton, J.-F., Zhang, X., Cheng, R.G.H., Klochkov, Y., Taufiq, M.F., Li, H.: Trustworthy llms: A survey and guideline for evaluating large language models' alignment. arXiv preprint arXiv:2308.05374 (2023)

[8] Majeed, A., Hwang, S.O.: Reliability issues of llms: Chatgpt a case study. IEEE Reliability Magazine (2024)

[9] Laskar, M.T.R., Chen, C., Tn, S.B., *et al.*: Are large language models reliable judges? a study on the factuality evaluation capabilities of llms. In: Proceedings of the Third Workshop on Natural Language Generation, Evaluation, and Metrics (GEM), pp. 310–316 (2023)

[10] Mirzaei, T., Amini, L., Esmaeilzadeh, P.: Clinician voices on ethics of llm integration in healthcare: A thematic analysis of ethical concerns and implications. BMC Medical Informatics and Decision Making **24**(1), 250 (2024)

[11] Wang, D., Zhang, S.: Large language models in medical and healthcare fields: applications, advances, and challenges. Artificial Intelligence Review **57**(11), 299 (2024)

[12] Haltaufderheide, J., Ranisch, R.: The ethics of chatgpt in medicine and healthcare: a systematic review on large language models (llms). NPJ digital medicine **7**(1), 183 (2024)

[13] Yoo, M.: How much should we trust llm-based measures for accounting and finance research? Available at SSRN (2024)

[14] Zhang, Z., Cao, Y., Liao, L.: Finbench: Benchmarking llms in complex financial problem solving and reasoning

[15] Bhattacharya, R., Aoun, M.A.: Using generative ai in finance, and the lack of emergent behavior in llms. Communications of the ACM **67**(8), 6–7 (2024)

[16] Wu, J., Yang, S., Zhan, R., Yuan, Y., Chao, L.S., Wong, D.F.: A survey on llm-generated text detection: Necessity, methods, and future directions. Computational Linguistics, 1–66 (2025)

[17] Minaee, S., Mikolov, T., Nikzad, N., Chenaghlu, M., Socher, R., Amatriain, X., Gao, J.: Large language models: A survey. arXiv preprint arXiv:2402.06196 (2024)

[18] Nouri, A.P., Hossain, M.S.: Difstorygen: Diffusion-based storytelling algorithm with distributed attention. In: 2024 IEEE International Conference on Big Data (BigData), pp. 761–768 (2024). IEEE

[19] Kumar, P.: Large language models (llms): survey, technical frameworks, and future challenges. Artificial Intelligence Review **57**(10), 260 (2024)

[20] Dam, S.K., Hong, C.S., Qiao, Y., Zhang, C.: A complete survey on llm-based ai chatbots. arXiv preprint arXiv:2406.16937 (2024)

[21] Nouri, A.P.: Dynamic storytelling algorithms using contextual aspects of a large language model. PhD thesis, The University of Texas at El Paso (2024)

[22] Patil, R., Gudivada, V.: A review of current trends, techniques, and challenges in large language models (llms). Applied Sciences **14**(5), 2074 (2024)

[23] Nouri, A., Hossain, M.S.: Corbs: a dynamic storytelling algorithm using a novel contextualization approach for documents utilizing bert features. Knowledge and Information Systems, 1–36 (2024)

[24] Ajwani, R., Javaji, S.R., Rudzicz, F., Zhu, Z.: Llm-generated black-box explanations can be adversarially helpful. arXiv preprint arXiv:2405.06800 (2024)

[25] Bhattacharjee, A., Moraffah, R., Garland, J., Liu, H.: Towards llm-guided causal explainability for black-box text classifiers. In: AAAI 2024 Workshop on Responsible Language Models, Vancouver, BC, Canada (2024)

[26] Du, X., Xiao, C., Li, S.: Haloscope: Harnessing unlabeled llm generations for hallucination detection. Advances in Neural Information Processing Systems **37**, 102948–102972 (2025)

[27] Perković, G., Drobnjak, A., Botički, I.: Hallucinations in llms: Understanding and addressing challenges. In: 2024 47th MIPRO ICT and Electronics Convention (MIPRO), pp. 2084–2088 (2024). IEEE

[28] Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Chen, Q., Peng, W., Feng, X., Qin, B., et al.: A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. ACM Transactions on Information Systems (2024)

[29] Jiang, L., Jiang, K., Chu, X., Gulati, S., Garg, P.: Hallucination detection in llm-enriched product listings. In: Proceedings of the Seventh Workshop on e-Commerce and NLP@ LREC-COLING 2024, pp. 29–39 (2024)

[30] Ji, Z., Yu, T., Xu, Y., Lee, N., Ishii, E., Fung, P.: Towards mitigating llm hallucination via self reflection. In: Findings of the Association for Computational Linguistics: EMNLP 2023, pp. 1827–1843 (2023)

[31] Su, W., Tang, Y., Ai, Q., Wang, C., Wu, Z., Liu, Y.: Mitigating entity-level hallucination in large language models. In: Proceedings of the 2024 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region, pp. 23–31 (2024)

[32] Gu, J., Jiang, X., Shi, Z., Tan, H., Zhai, X., Xu, C., Li, W., Shen, Y., Ma, S., Liu, H., et al.: A survey on llm-as-a-judge. arXiv preprint arXiv:2411.15594 (2024)

[33] Li, D., Jiang, B., Huang, L., Beigi, A., Zhao, C., Tan, Z., Bhattacharjee, A., Jiang, Y., Chen, C., Wu, T., et al.: From generation to judgment: Opportunities and challenges of llm-as-a-judge. arXiv preprint arXiv:2411.16594 (2024)

[34] Curran, C., Neehal, N., Murugesan, K., Bennett, K.P.: Examining trustworthiness of llm-as-a-judge systems in a clinical trial design benchmark. In: 2024 IEEE International Conference on Big Data (BigData), pp. 4627–4631 (2024). IEEE

[35] Beigi, A., Tan, Z., Mudiam, N., Chen, C., Shu, K., Liu, H.: Model attribution in llm-generated disinformation: A domain generalization approach with supervised contrastive learning. In: 2024 IEEE 11th International Conference on Data Science and Advanced Analytics (DSAA), pp. 1–10 (2024). IEEE

[36] Pal, A., Umapathi, L.K., Sankarasubbu, M.: Med-halt: Medical domain hallucination test for large language models. arXiv preprint arXiv:2307.15343 (2023)

[37] Tonmoy, S., Zaman, S., Jain, V., Rani, A., Rawte, V., Chadha, A., Das, A.: A

comprehensive survey of hallucination mitigation techniques in large language models. arXiv preprint arXiv:2401.01313 (2024)

[38] Xu, Z., Jain, S., Kankanhalli, M.: Hallucination is inevitable: An innate limitation of large language models. arXiv preprint arXiv:2401.11817 (2024)

[39] Orgad, H., Toker, M., Gekhman, Z., Reichart, R., Szpektor, I., Kotek, H., Belinkov, Y.: Llms know more than they show: On the intrinsic representation of llm hallucinations. arXiv preprint arXiv:2410.02707 (2024)

[40] Lin, C.-Y.: Rouge: A package for automatic evaluation of summaries. In: Text Summarization Branches Out, pp. 74–81 (2004)

[41] Papineni, K., Roukos, S., Ward, T., Zhu, W.-J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pp. 311–318 (2002)

[42] Dai, S., Xu, C., Xu, S., Pang, L., Dong, Z., Xu, J.: Bias and unfairness in information retrieval systems: New challenges in the llm era. In: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 6437–6447 (2024)

[43] Farquhar, S., Kossen, J., Kuhn, L., Gal, Y.: Detecting hallucinations in large language models using semantic entropy. Nature **630**(8017), 625–630 (2024)

[44] Kang, H., Blevins, T., Zettlemoyer, L.: Comparing hallucination detection metrics for multilingual generation. arXiv preprint arXiv:2402.10496 (2024)

[45] Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q., Artzi, Y.: Bertscore: Evaluating text generation with bert. arXiv preprint arXiv:1904.09675 (2019)

[46] Galitsky, B.: Truth-o-meter: Collaborating with llm in fighting its hallucinations. In: Interdependent Human-Machine Teams, pp. 175–210. Elsevier, ??? (2025)

[47] Reddy, G.P., Kumar, Y.P., Prakash, K.P.: Hallucinations in large language models (llms). In: 2024 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream), pp. 1–6 (2024). IEEE

[48] Rateike, M., Cintas, C., Wamburu, J., Akumu, T., Speakman, S.: Weakly supervised detection of hallucinations in llm activations. arXiv preprint arXiv:2312.02798 (2023)

[49] Yao, J.-Y., Ning, K.-P., Liu, Z.-H., Ning, M.-N., Liu, Y.-Y., Yuan, L.: Llm lies: Hallucinations are not bugs, but features as adversarial examples. arXiv preprint arXiv:2310.01469 (2023)

[50] Yu, X., Cheng, H., Liu, X., Roth, D., Gao, J.: Automatic hallucination assessment for aligned large language models via transferable adversarial attacks. arXiv

preprint arXiv:2310.12516 (2023)

[51] Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., et al.: Judging llm-as-a-judge with mt-bench and chatbot arena. Advances in Neural Information Processing Systems **36** (2024)

[52] Kalra, N., Tang, L.: Verdict: A library for compound llm judge systems

[53] Song, J., Wang, X., Zhu, J., Wu, Y., Cheng, X., Zhong, R., Niu, C.: Rag-hat: A hallucination-aware tuning pipeline for llm in retrieval-augmented generation. In: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track, pp. 1548–1558 (2024)

[54] Hu, X., Zhang, Y., Peng, R., Zhang, H., Wu, C., Chen, G., Zhao, J.: Embedding and gradient say wrong: A white-box method for hallucination detection. In: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pp. 1950–1959 (2024)

[55] Lee, H., Park, K.-H., Byun, H., Yeom, J., Kim, J., Park, G.-M., Song, K.: Ced: Comparing embedding differences for detecting out-of-distribution and hallucinated text. In: Findings of the Association for Computational Linguistics: EMNLP 2024, pp. 14866–14882 (2024)

[56] Malin, B., Kalganova, T., Boulgouris, N.: A review of faithfulness metrics for hallucination assessment in large language models. arXiv preprint arXiv:2501.00269 (2024)

[57] Hasegawa, R., Ichise, R.: Cokglm: Detecting hallucinations generated by large language models via knowledge graph verification. In: International Knowledge Graph and Semantic Web Conference, pp. 212–224 (2024). Springer

[58] Sansford, H., Richardson, N., Maretic, H.P., Saada, J.N.: Grapheval: A knowledge-graph based llm hallucination evaluation framework. arXiv preprint arXiv:2407.10793 (2024)

[59] Guan, X., Liu, Y., Lin, H., Lu, Y., He, B., Han, X., Sun, L.: Mitigating large language model hallucinations via autonomous knowledge graph-based retrofitting. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, pp. 18126–18134 (2024)

[60] Jin, H., Zhang, Y., Meng, D., Wang, J., Tan, J.: A comprehensive survey on process-oriented automatic text summarization with exploration of llm-based methods. arXiv preprint arXiv:2403.02901 (2024)

[61] Fang, J., Liu, C.-T., Kim, J., Bhedaru, Y., Liu, E., Singh, N., Lipka, N., Mathur, P., Ahmed, N.K., Dernoncourt, F., et al.: Multi-llm text summarization. arXiv preprint arXiv:2412.15487 (2024)