AI Hallucination Detector

A major project report submitted in partial fulfillment of the requirement for the award of degree of

Bachelor of Technology

in

Computer Science & Engineering

Submitted by

Arpita Rani (221030055)

Anand Chaudhary (221030123)

Rishal Rana (221030004)

Arnav Sharma (221030059)

Under the guidance & supervision of

Prof. Dr. Vivek Kumar Sehgal

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology, Waknaghat, Solan - 173234 (India)

December 2025

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS, NOMENCLATURE

Abbreviation/Symbol	**Full Form**
AI	Artificial Intelligence
LLM	Large Language Model
SRL	Semantic Role Labeling
RAG	Retrieval- Augmented Generation
BLEU	Bilingual Evaluation Understudy
ROUGE	Recall-Oriented Understudy for Gisting Evaluation
ACL	Association for Computational Linguistics
GPT	Generative Pre-trained Transformer
NLP	Natural Language Processing
QA	Question Answering
PDF	Probability Density Function

ABSTRACT

Artificial intelligence system which mainly includes various multimodal recently got a lot of attention because of their outputs , the reasoning that they give , the analysis that they made and so on. But we haven't gone through one thing in this which is acting as a very big problem which affects the output and affects the trust of all of us which is nothing but hallucinations. But why it is happening? That's the main question to all of us. What exactly is wrong in all AI Models that they are hallucinating. Think about it. It's a very big problem in sectors such as health,law and finance especially, one term changes the whole situation, so correct information is really important.

Why have we taken this project? To understand why this is happening, what is the root cause of all this , how to detect the hallucinations and how we can reduce it. Though their are many models and applications that are serving this thing but we have in our mind that how can we apply some unusual things on the model. So we are trying to apply a model which can automatically check things , applying attention based features, taxonomy based evaluation not only for text but also for videos images and audios.

So what we have seen till now that the datasets that models are using is still facing some kind of issues which lacks reasoning, proper understanding and a lot of instructions . We have seen that the ways that are used to reduce hallucinations till now helps us but not completely and not on a satisfactory level. So to make models reliable we have to apply a lot of changes to improve the reliability and accessibility to the understanding, reasoning, set of instructions, datasets etc.

By this project we are looking firstly into the problems that comes in the datasets like data leakage, cost problems and various other problems. We need to make a model which helps human to trust Artificial intelligence much more and to become trustworthy so that everyone can apply AI in real world applications specially in the areas of law finance and healthcare.

Chapter 1: INTRODUCTION

1.1 Introduction

Artificial intelligence system which mainly includes large language models and various other models have gotten popular so much because of the solutions that they are genrating , the reasoning that they are giving , the analysis that they are making and so on.

But all these abilities are not making full impact on all the real world applications because of one serious problem of hallucination.

- What is Hallucination? When AI generates an answer which looks very convincing to us but it is not actually true.
- Main reason for that is that they generate responses on the basis of the pattern that they are trained on , not the actual facts.
- That clearly means that if they are not completely aware about the actual facts they tend to make mistakes.

If we try to understand real world applications, hallucination is a big problem in sectors like:

- Medical/ Health – Probability of getting wrong answer.
- Law – False Allegations.
- Finance – Wrong Transaction Statements

1.2 Problem Statement

While LLMs are very good at generating natural and human-like responses their habit of creating hallucinated content makes them less reliable and less trustworthy.

- Current evaluation metrics like BLEU and ROUGE mainly check how similar the generated text is to the reference text in terms of language. They don't check if the information is actually correct, which means a text can get a high score even if it is factually wrong.

- The existing methods to reduce hallucination are also not good enough because:
  - Many of them work only for a specific field (for example, they might work in medicine but fail in other areas).
  - Some of them need too much computing power so they can't be used in real-time applications.
  - Others are not clear enough they give a score but don't explain why something is wrong.

So, there is a clear need for a hallucination detection framework that is:

- Effective → can correctly find factual mistakes.
- Interpretable → can explain the reason, not just give a number.

If system like we are discussing comes in picture , it will be a boom for AI Industry and it will be really beneficial to real World.

1.3 Objectives

Let's discuss what are the main objectives of this project:

Main objection is to understand why hallucination occures in Large Language Models(LLMs).

Conversation should include that what is hallucinations and why it occurs in various models. Thenit should includes how hallucinations occur and how it checks the tokens and Pattern of the data that is provided to them, so let us see where things can go wrong in the dataset:

- Unclear or confusing user questions.
- Rare or new situations that the model has never seen before.
- The model being too confident even when it is unsure.

● By studying these reasons, the project will create a strong base for detecting and reducing hallucinations in a better way.

2. **To check the detection and methods through a detailed study.**
   - First we will start by reading and researching all the research papers on this topic to understand the problem and its solution well.
   - This includes checking different methods like:
     - Detection approaches such as black-box scorers white-box scorers, retrieval augmented generation (RAG) and LLM-as-a-judge techniques. Strategies are using various models to check the coding and encoding schemas of various datasets.
     - This review will help us to determine the gaps between various models , various datasets and it will help us to understand that where we have to pick up to start for the project.
     - This step will make sure that the project builds on the latest research instead of repeating what has already been done.

3. **To create and propose an AI hallucination detection framework that checks factual correctness in generated text.**
   - Based on what is learned from the research, the project will design a new detection framework that has a modular structure.
   - The main parts of this framework will include:
     - **Fact Extraction Module:** Changes sentences into fact-based structures like who did what where and when.
     - **Scoring Modules:** Uses different strategies like black-box, white-box, LLM-as-a-judge and combined(ensemble)scoring methods.
     - **Hallucination Detection Module:** Combines all the scores to mark false or unsupported statements.
     - **Output Module:** Gives clear and easy-to-understand reports showing which parts of the text have hallucinations

4.To provide ideas to make AI more reliable and trustworthy

● In the end, the project will share useful suggestions and best practices for researchers, developers, and people working in the AI industry.

● Main moto is to provide a better future AI work so that it becomes safer , better and trustworthy.

1.3 Significance And Motivation Of The Project Work

The whole moto that comes under the growing of Artificial Intelligence(AI) is that in this accuracy should beat the top notch.

We can see the importance by the points given:

- **Trustworthiness:** Give more reliability to the models by reducing false or made-up responses.
- **Practical Impact:** Helping to use LLMs safely in important areas like healthcare, law, finance,and education.
- **Contribution of the Research**: Provide a proper framework for finding hallucinations that can serve as the basis for further work toward making AI safer**.**

The development of this project is not only important from a research point of view but also has great social impact since it focuses on solving one of the biggest problems in today's AI usage.

1.4 Organization Of Project Report

The rest of this project report is arranged as follows:

- **Chapter 2:** Literature Review This presents the research work available on hallucinations in LLMs, their detection methods, and ways of reducing them**.**
-
- **Chapter 3: System Development** – Explains the project's design and overall architecture of the project.

CHAPTER 2: LITERATURE SURVEY

2.1 Overview Of Relevant Literature

Artificial Intelligence (AI) and Natural Language Processing (NLP) have grown very fast during the past decade-with large language models such as GPT, BERT, and LLaMA showing top-level performance in many areas. But even with all these great results, one The following common problem keeps cropping up: hallucination, where LLMs create text that looks Correct and fluent but is actually factually wrong. This section gives an overview of recent Research on hallucinations from 2019 to 2024: how to detect them, how to reduce them..

Several studies have tried to clearly define and understand various types of hallucinations. Huang et al. (2023) created a classification system for hallucinations in LLMs, dividing them into two main types: **factuality hallucinations,** which go against real-world facts, and **faithfulness hallucinations** which are inconsistent with the user's input or the model's own logic. Their study also explained how hallucinations can happen during the data, training, and inference stages and mentioned evaluation benchmarks like **TruthfulQA** and **HaluEval**.

Zhang et al. (2023) pointed out, in their survey entitled *"Siren's Song in the AI Ocean"*, that it's not just about the poor quality of data, but about how the model generates the hallucinations, text for example via decoding methods like temperature sampling. They also discussed the balance between creativity and factual accuracy and suggested techniques such as **Retrieval-Augmented Generation (RAG)**, **consistency checking**, and **uncertainty estimation** as effective ways to reduce hallucinations.

Other important contributions include:

- **TruthfulQA (Lin et al., 2022):** A benchmark that measures hallucination levels in different areas, showing that even top LLMs often hallucinate in law, science and medical topics.

● **SelfCheckGPT (Manakul et al., 2023):** A method that checks for hallucinations by comparing multiple outputs from the same model to see if the answers stay consistent.

● **REALTIMEQA (Kasai et al., 2023):** A dataset made for real-time or time-sensitive questions, showing that hallucinations increase when the information change over time.

● **Hallucination in Multimodal Models (Bai et al., 2024):** Expanded the study of hallucinations to vision-language models, where mistakes can happen both in image understanding and text generation.

All of these studies together give a full picture of what hallucinations are why they matter, and how researchers are working toward better ways to detect and reduce them for more trustworthy AI systems.

Table 1: Literature Survey

S.No	Author & Paper Title	Journal/ Conference	Tools/ Techniques/ Dataset	Key Findings/ Results	Limitations/ Gaps Identified
1	Brahmaleen Kaur Sidhu, "Hallucinations in Artificial Intelligence: Origins, Detection, and Mitigation" [SR241229170309]	IJSR (2025)	Review; SelfCheckGPT, ChatProtect, GPTScore, RHO, NPH, RAG, MixCL, HERMAN; datasets incl. WikiBio, OpenDialKG	Categorizes AI hallucinations, surveys detection/mitigation models, urges multi-metho d approaches	Detection tools lack context focus; output quality gaps; factual alignment remains challenging
2	Zhiyang Chen et al., "Mitigating Hallucination in Visual Language Models with Visual Supervision" [arXiv:2311.16479]	arXiv (2023)	Visual annotation (PSG, SAM), Mask prediction, RAH-Bench	Fine-grained annotation, mask prediction loss, and visual supervision help LVLMs reduce three types of	Data format coupling; limited dataset size; method needs generalizatio n across annotation formats

				hallucination and outperform baseline models	
3	Dylan Bouchard et al., "UQLM: Uncertainty Quantification for Language Models" [arXiv:2507.06196v1]	arXiv (2025)	Python package 'uqlm', UQ-based scoring methods (Black-Box, White-Box, LLM-as-Judge), multiple existing benchmarks	Provides library for uncertainty-based hallucination detection at generation time; democratizes advanced quantification for LLM outputs	Limited technique scope in prior tools; adoption and ease-of-use for non-experts remains; external fact-checking still a challenge
4	Adam Tauman Kalai et al., "Why Language Models Hallucinate" [OpenAI/Georgia Tech, Sep 2025]	Preprint (2025)	Theoretical analysis, meta-evaluation of LLM benchmarks	Hallucinations arise from statistical pressures in training; binary evaluations reward guessing over uncertainty	Benchmarks penalize uncertainty; socio-technical fixes needed in evaluation/grading; open questions on uncertainty signaling
5	Zhang et al., “Siren’s Song in the AI Ocean: A Survey on Hallucination in Large Language Models,” 2023	*arXiv preprint* (2023).	Literature review across detection & mitigation strategies.	Showed hallucinations are systemic; reviewed detection (fact-checking, NLI, uncertainty) and mitigation (RAG, consistency, instruction tuning).	Conceptual; lacks empirical evaluation; limited to text-based LLMs

6	Chaoyou Fu et al. MME: A Comprehensive Evaluation Benchmark for Multimodal Large Language Models [Fu et al., arXiv:2306.13394]	arXiv Preprint (2024)	**Technique:** MME benchmark with manual yes/no instruction-answer pairs. **Models:** 30 MLLMs (e.g., GPT-4V, LLaVA). **Metrics:** Accuracy, Accuracy+.	Benchmarked perception (existence, count, color, position, OCR, etc.) and cognition (reasoning, calculation, translation, code) abilities for 30 models. Identified persistent gaps: failure to follow instructions, perception errors, object hallucination, and reasoning limitations.	Models struggle with concise instructions, object perception, and universal robust reasoning. Data leakage risk and challenge of instruction generalization. Not all cognitive tasks achieved high accuracy; further benchmarking needed.

7	Xu et al. *Hallucination is Inevitable*	ArXiv 2024	Theoretical proof using learning theory & diagonalizat i on. Defines a formal world with computable LLMs	Proves hallucinatio n is mathematica lly inevitable for any computable LLM, regardless of architecture or training data. Identifies hallucinatio n-prone problem classes	Analysis is theoretical ; doesn't quantify real-world hallucinatio n rates. Assumes deterministi c ground truth, limiting probabilistic insights. Focuses on inherent limits,
8	Maleki et al. *AI Hallucinations: A Misnomer Worth*	Preprint (e.g., ArXiv) 2024	Systematic review of 14 databases	Highlights inconsistent and often	Manual review may miss newer

	Clarifying		from 2013–2023. Manually extracted and analyzed 333 definitions of "AI hallucinatio n."	contradictor y definitions across domains. Suggests alternative terms (e.g., fabrication, stochastic parroting). Calls for more precision	definitions. Focuses on terminology, not technical solutions. Limited to English-lang uage papers. Does not propose a unified taxonomy.

9	Pranab Sahoo et al. A Comprehensive Survey of Hallucination in Large Language, Image, Video and Audio Foundation Models	arXiv Preprint (2024)	**Technique:** Systematic taxonomy and review of detection/mitigation strategies	Provides a unified framework and synthesizes detection/mitigation strategies.	Scope is broad, not deep. Cuts off at May 2024, missing recent works.
10	Chaoyou Fu et al. MME: A Comprehensive Evaluation Benchmark for Multimodal Large Language Models [Fu et al., arXiv:2306.13394]	arXiv Preprint (2024)	**Technique:** MME benchmark with manual yes/no instruction-answer pairs. **Models:** 30 MLLMs (e.g., GPT-4V, LLaVA). **Metrics:** Accuracy, Accuracy+.	Benchmarked perception (existence, count, color, position, OCR, etc.) and cognition (reasoning, calculation, translation, code) abilities for 30 models. Identified persistent gaps: failure to follow instructions, perception errors, object hallucination, and reasoning limitations.	Models struggle with concise instructions, object perception, and universal robust reasoning. Data leakage risk and challenge of instruction generalization. Not all cognitive tasks achieved high accuracy; further benchmarking needed.

11	Cheng Niu1 et al. RAGTruth: A Hallucination Corpus for Developing Trustworthy Retrieval-Augmented Language Models	arXiv Preprint 2024	Used **GPT-3.5, GPT-4, Llama-2, Mistral-7B** to generate responses, Detection methods: **Prompt-based (GPT), SelfCheckGPT, LMvLM, Fine-tuned Llama-2-13B, Data set RAGTruth**	Data-to-text had the most hallucinations (68%); GPT-4 the least, Mistral-7B the most. Fine-tuned Llama-2-13B reached 78.7% F1 and cut hallucinations by 50–63%	**Limitations:** Focused only on RAG and three tasks, span-level detection weak, annotations subjective, results may not generalize. **Gaps:** Need bigger diverse datasets, better subtle hallucination handling, cross-task/multilingual tests
12	Junyang Wang et al. AMBER: An LLM-free Multi-dimensional Benchmark for MLLMs Hallucination Evaluation	arXiv Preprint 2024	Proposed **AMBER**, an **LLM-free benchmark** for hallucination evaluation in Multi-modal LLMs (MLLMs), Dataset- **MSCOCO test set**	GPT-4V performed best with the least hallucinations and highest AMBER Score, followed by Qwen-VL and InstructBLIP. However, all models still showed hallucinations, especially in **attribute** and **relation** cases	AMBER only evaluates **attribute and relation hallucinations** in discriminative tasks, not generative ones. Object extraction errors may occur (e.g., “orange” as noun vs color)

13	Balaji Padmanabhan et al. AI Hallucinations: A Misnomer Worth Clarifying	arXiv 2024	Systematic literature review (14 databases, 333 papers), manual screening, categorization by domain	No consistent definition of "AI hallucination"; term used inconsistently; alternatives like *confabulation, fabrication, misinformation*;	Focus on definitions not solutions.Need unified taxonomy, standardized non-stigmatizing terms

14	Varun Magesh et al. Hallucination-Free ? Assessing the Reliability of Leading AI Legal Research Tools	Journal of Empirical Legal Studies (2025)	Empirical tests on Lexis+ AI, Westlaw, Practical Law AI, GPT-4 with 200+ legal queries, hand-coded evaluation.	All still hallucinate (Lexis+ ~17%, Westlaw ~33%, Practical Law high incompleteness, GPT-4 worst); RAG helps but not a fix.	Proprietary black boxes, small dataset, US-law only, snapshot in time.Need bigger cross-jurisdiction studies, better benchmarks, lawyer-side mitigation, long-term ethical analysis.

15	Fan, Aumiller, and Gertz, “Evaluating Factual Consistency of Texts with Semantic Role Labeling,” 2023	ACL 2023 Proceedings	Semantic Role Labeling (SRL); datasets: QAGS (CNN/XSUM), SummEval.	Proposed SRLScore (reference-free factuality metric); achieved high correlation with human judgment; interpretable	Slower than LM-based methods; only tested in English; occasional mismatch with human evaluations
16	Guliyev and Özer, “On the ‘Hallucinations’ of Artificial Intelligence and Their Terminologies, ” 2024.	*Cureus* (2024).	Conceptual analysis of terminology.	Argues “hallucination” is metaphorical, not literal; proposes using terms like fabrication/confabulation instead.	Theoretical only; no technical solutions
17	Huang et al., “A Survey on Hallucination in Large Language Models,” 2023.	*ACM Computing Surveys* (2024).	Literature survey of LLM hallucination studies	Provided taxonomy of hallucinations (factual vs. faithfulness); identified causes (data, model, decoding); reviewed benchmarks like TruthfulQA and HaluEval	Survey only; no new method proposed; limited multimodal/multilingual coverage.

18	Santosh S. Vempala et al.	arXiv Preprint 2025	Computational learning theory, binary classification reduction, error analysis, benchmark evaluation review, confidence-targeted evaluation	Hallucinations are statistically inevitable; caused by arbitrary facts, poor models, and evaluation design; benchmarks reward guessing over "I don't know."	Mostly theoretical, simplified binary framework, focused on factual QA, little empirical validation.Need fine-grained taxonomy, large-scale empirical tests, domain-specific study, and better evaluation metrics

19	Jeffrey P. Bigham et al.	arXiv preprint / Presented at Human-Centered Machine Learning Workshop (2024); Scheduled at ACL (Vienna, 2025)	MMAU Benchmark PLUM (Pipeline for Learning User Conversations in LLMs) External Validation Tools	MMAU provides more comprehensive and interpretable evaluation compared to older benchmarks. PLUM improves personalization by leveraging conversational history, beyond static user preferences. External validation sometimes improves annotation quality and reduces biases in LLM-as-a-Judge setups.	External validation tools improved results only *"often, but not always"* Personalization in PLUM still limited to extracted Q–A pairs Apple strong in research output, but practical deployment into consumer-facing AI products lags behind competitors.
20	Alkaissi and McFarlane, "Artificial Hallucinations in ChatGPT: Implications in Scientific Writing," 2023.	*Cureus Journal of Medical Science* (2023)	Tested ChatGPT on medical topics (Homocystinuria, Pompe disease).	Showed ChatGPT fabricates citations and unsupported claims; highlights risks in medical/scientific domains	Case-based only; not systematic; no general detection method proposed.

2.2 Key Gaps In The Literature

1. Lack of Benchmarks

   Right now the benchmarks used for detecting hallucinations are very narrow and depend on specific tasks For example:

   ● In healthcare, datasets focus on checking if the model gives factually correct diagnoses.

   ● In news, datasets check whether the summaries match the original articles.

   But there is no single benchmark that works across different areas like health, law, education, finance, general Q&A, or creative writing. In simple words we can say that, without a common & standardized test it can creates difficulty to compare different methods with a proper systematic way .

2. Overreliance on Model-Based Metrics

   Most available evaluation methods today, including BARTScore or QA-based evaluators, rely on. It allows other large models to decide if an answer is a hallucination.

   - BARTScore relies on a pre-trained model to assess the closeness of a generated
   - Problem: These "judge models" can have their own biases and mistakes, which
   - makes their results less trustworthy.
   - Also, they act like black boxes they can mark a sentence as hallucinated but can't clearly explain why.

   This creates a circular problem: using one imperfect model in order to evaluate another one.

3. Limited Real-Time Applicability

   Some methods like RAG (Retrieval-Augmented Generation) or SelfCheckGPT lot of computing power.

   ● RAG constantly retrieves information from external databases.

   ● SelfCheckGPT makes multiple versions of an answer to compare and check for consistency.

These techniques are too heavy to be used in real-time applications such as:

- Customer support chatbots
- Financial trading assistants
- Emergency healthcare bots

Because of this, there is still a big gap between what works in research and what can actually be used in real-world systems.

4. Insufficient Multilingual and Multimodal Research
   - Most research has focused only on English-language hallucinations.
   - Very few studies have tested hallucinations in multilingual LLMs like those working in Hindi Spanish or Mandarin.
   - Even fewer have looked at multimodal models that handle text, images, audio, or video.

   For instance, a multimodal model might describe an image incorrectly or create a wrong caption for a video but this kind of error has not been studied much. As a result, global and multimodal AI systems remain underexplored.

5. Inadequate Handling of Uncertainity
   LLMs often sound very confident, even when they are completely wrong.
   - Example: If a model doesn't actually know a fact, instead of saying "I don't know," it may create a detailed but false answer.
   - Only a few approaches try to make models show uncertainty or give probabilistic answers.

   Building LLMs that can admit when they are unsure would greatly reduce risks, especially in sensitive areas like medicine, law, and finance.

6. Trade-off between Creativity and Accuracy
   - One way to lower hallucinations is to reduce randomness during text generation (for example lowering temperature or Top-k sampling).
   - But doing this also makes the model's responses repetitive, predictable, and less creative.

The things come out is balancing, its play's major role in every area of filed special to the creative where we need brainstorming, storytelling, marketing execution, content creation.

So, researchers must find the right balance keeping outputs accurate without completely losing the model's creativity.

CHAPTER 3: SYSTEM DEVELOPMENT

3.1 Requirements And Analysis

The design and development of the AI Hallucination Detector need to look at both the **functional requirements** (what the system should actually do) and the **non-functional requirements** (how well it should perform and behave). Based on the project scope and what has been learned from the research the requirements are written below:

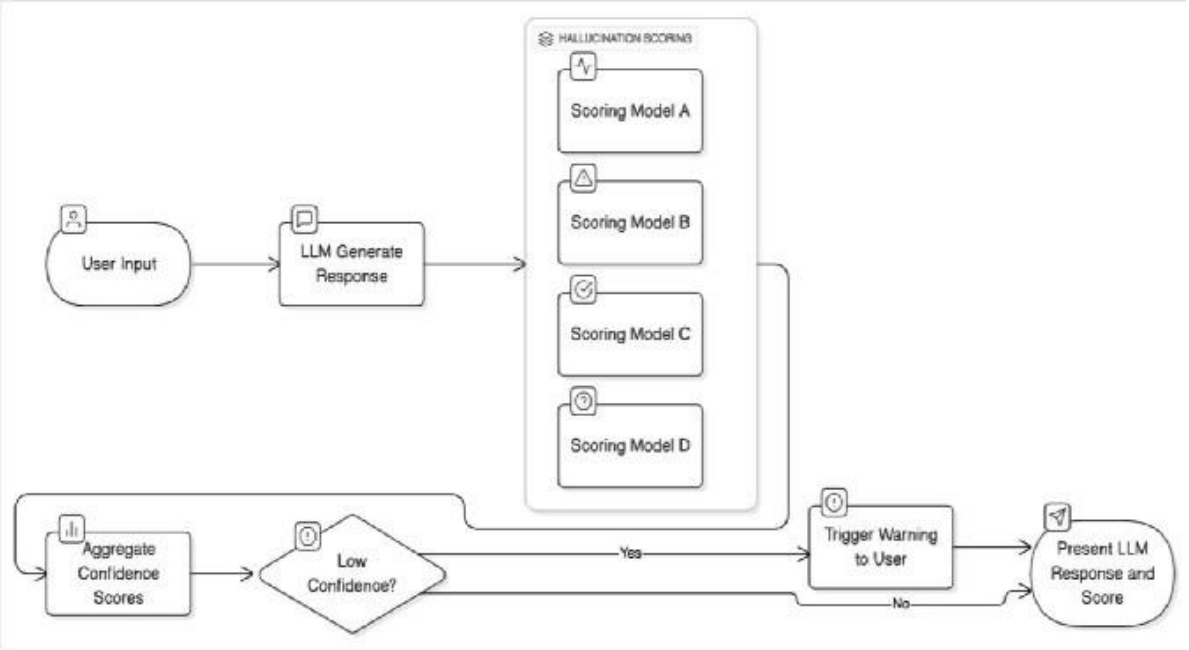

Figure 3.1 : Functional Requirements of the AI Hallucination Detector

3.1.1 Functional Requirements

1. **Input Handling**
   - The system should be able to take AI-generated outputs from models like GPT, BERT or LLaMA.
   - It should also allow users to give optional reference text or datasets that can be used to check facts.
2. **Fact Extraction**
   - We should use various Natural Language Processing techniques to find the actual facts about the object.
3. **Fact Matching**
   - Next one should compare the reference and generated text so that it matches the

knowledge base.

4. **Hallucination Detection**.
   ○ Then find the factual score for this.

5. **Evaluation and Benchmarking**
   ○ Give clear reports that shows the generated text that contains hallucination and where we should remove this.

3.1.2 Non-Functional Requirements

1. **Scalable:** Works well with both short and long answers.
2. **Interoperability:** Able to connect with different Large Language Models.
3. **Interpretability:** Easy to understand..
4. **Efficiency:** Optimized enough to handle real time applications.
5. **Domain Adaptability:** Works in every domain.
6. **Reliability:** Should be enough reliable..

3.2 Project Design And Architecture

The system proposed is designed in such a modular way,so that it is flexile and scalable.Multiple detection methods are combined together such as black-box,white-box and ensemble scorers to make the result more accurate

3.2.1 System Workflow

The overall working process of the AI Hallucination Detector can be explained in the following steps:

1. **Input Module**
   - Input : AI-generated text
   - Can also obtain optional reference documents for verification.

2. **Fact Extraction Module**
   - Decomposes the sentence using Semantic Role Labeling and parsing**.**
   - Converts the text into format of fact tuples such who, did what, to whom, where, and when.

3. **Scoring Modules**
   - **Black-Box Scorers:** Determine consistency of output with reference by measuring overlaps..
   - **White-Box Scorers:** Use the model's own statistics about word probabilities to identify unlikely or low-confidence outputs..
   - **LLM-as-a-Judge:** A different model reviews the generated text and evaluates the correctness.
   - **Ensemble Scorers:** Combine all scorer outputs of individual models to obtain more reliable score..

4. **Hallucination Detection Module**
   - Add different scores that we get from scorers..
   - Find facts (actual).
   - Find the total hallucination score with explanation.

5. **Output and Visualization Module**
   - Making an easy understandable report..
   - Detect and highlight where we are finding hallucination.
   - Plus provide the result by considering various scorers..

3.2.2 System Architecture

System Architecture of our project is:

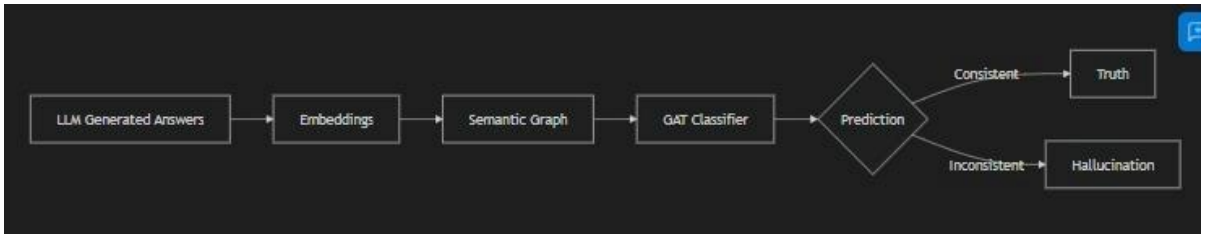

Figure 3.2: System Architecture of the Proposed AI Hallucination Detector

3.2.3 Design Characteristics

1. Multi-Scorer Strategy
   - We can use various scorer models to generate answers..
   - Example:
   - Like we have Semantic similarity that checks that the meaning of generated text matches the reference text or not.

   - Benefit: If one check is not getting an answer the other will.

2. Explainability
   - Latest systems are basically black boxes.They will give out a score but will never explain how they reached this conclusion which then makes it difficult to understand or trust the score.
   - With our project, the system clearly gives where the factual issue is instead of just giving out scores.
   - Example: lets say the model claims “Covid-19 started in 2018” , the system will show that part and show that the correct information says the beginning of COVID-19 is in 2019.
   - Advantages:Transparency and also both users and developers trust it more which is necessary in fields like healthcare and legal matters.

3. Extensibility
   - The system is designed to easily integrate with external data sources.
   - such as Wikipedia articles, academic papers, or niche-specific datasets.
   - Example:-
     - For detecting medical hallucinations, it can utilize PubMed or WHO databases.
     - For finance-focused identification, it can link to stock market or economic data sources.
   - Advantage: This maintains flexibility in the framework and enables its application across various sectors without the necessity of reconstructing the entire system

4. Robustness
   - If we rely on only one detection method, it’s basically a gamble — the moment it fails, hallucinations can easily slip through without anyone noticing.

   - That’s why combining different evaluators and bringing in external sources makes the whole system much more trustworthy. It doesn’t crash just because one part messes up.

   - For example, if a semantic similarity checker can’t catch something because the model rephrased it too cleverly, the factual accuracy checker can still flag the mistake.
   - In short, using multiple layers makes the system a lot more stable, consistent, and reliable across all kinds of inputs and situations.

3.3 Data Preparation

Our first point in making this project is to create a good dataset which should be enough reliable. Hallucination depends mainly on various different answers and then comparing that answers with the questions and the prompt , so the database that we require would be very systematic..

A list of prompt is given to the script and then the prompts run with some extra information we have . The script also helps us to generate answers or we can say responses that helps to find difference in the model.

tThe responses that we have should be saved in files like .csv or .json under the directory. We should organise the raw output into clean outputs. It also performs basic data cleaning wherever necessary.

By the end of this step, each prompt will have a small group of different responses linked to it, which becomes the base for graph construction and later model training.

Overall, the dataset prepared at this stage contains a mix of correct, partially correct, and hallucinated responses which is very important for training a system that can accurately tell the difference between them

```python
import pandas as pd
import json

# Load Generated Data
df_wc = pd.read_csv("data/generated/with_context.csv")
print("With Context Data:")
print(df_wc.head())

# Load Sampled JSON
with open("data/sampled_data.json", "r") as f:
    sampled_data = [json.loads(line) for line in f]
print(f"\nLoaded {len(sampled_data)} sampled questions.")
```

Fig 3.3.1 : Sample Data Loading Code

Fig. 3.3.2: Dataset Snippet

3.4 Implementation

This project is mainly implemented in **Python** with **Jupyter notebooks** used for experiments and an **environment.yml** file to manage all dependencies. The whole implementation is divided into different modules, where each module handles a specific step of the hallucination detection process.

a) Tools and Libraries

The project makes use of the following tools and technologies:

- **Python** along with common data-processing libraries.
- **Optional GPU support using CUDA** for faster computations.
- **Embedding models** to create vector representations of text.

Graph processing tools and a **Graph Attention Network (GAT)** model for analyzing relationships between responses.

- **Baseline models** used for performance comparison.

The codebase is divided into distinct folders for running baseline methods, creating graphs, storing trained model weights, and storing datasets.

b) Overview of PIpeline

The system operates according to a four-step process:

1. **Collect multiple answers for each prompt.**
   Using the Scripting Files (i.e. , generation.py) to create responses and determining what parameters will be used for sampling (i.e. random seed, temperature, etc.) to enable the creation of variations of those responses.
2. **Convert each answer into an embedding.**
   All of the generated responses will become vectors so that similarity can be calculated between responses.
3. **Build a similarity-based graph.**
   The response vector will become a node in the graph. Graph edges will be created between nodes of any two response vectors that have an identified high degree of semantic similarity. This association creates a graph that illustrates how close or far apart different responses are to each other.

4. **Train and evaluate a Graph Attention Network(GAT).**
   The GAT models (Algorithm for Ticket Allocation) take advantage of the connections made in GAT to learn the associations between response results to identify which responses are grounded and hallucinated.. The repository includes tools for model training, evaluation, and graph visualization to help understand how the model performs.

c) Illustrative Implementation Snippet

Below is a pseudocode-style example showing how the detection process works internally. In the actual codebase, these steps are divided into clear, separate modules to make experimentation easier and better organized.

d) Baseline Methods

The **baseline** folder includes simpler methods that do not use graph-based processing. These methods evaluate each response independently and are used to compare how much better the graph-based model performs in detecting hallucinations.

Embedding Code:

```
from sentence_transformers import SentenceTransformer
model = SentenceTransformer("bert-base-uncased")
texts = ["Answer 1", "Answer 2"]
emb = model.encode(texts)
print(f"Embedding shape: {emb.shape}")
```

Fig. 3.4.1: Embedding Code

Graph Construction Code:

```
import numpy as np
from sklearn.metrics.pairwise import cosine_similarity
import matplotlib.pyplot as plt

# Dummy embeddings
embeddings = np.random.rand(10, 128)

# Compute Similarity
sim = cosine_similarity(embeddings)

# Create Edges
threshold = 0.85
edges = np.argwhere(sim > threshold)
# Remove self-loops
edges = edges[edges[:, 0] != edges[:, 1]]

print(f"Created {len(edges)} edges.")

# Plot Degree Distribution
degrees = np.bincount(edges[:, 0], minlength=10)
plt.figure()
```

Fig. 3.4.2: Graph Construction Code

Training Loop Snippet:

```
import torch.optim as optim

model = GATNet(128, 32, 4)
optimizer = optim.Adam(model.parameters(), lr=0.001)
criterion = torch.nn.CrossEntropyLoss()

train_losses = []

model.train()
for epoch in range(100): # Dummy loop
    optimizer.zero_grad()
    out = model(graph.x, graph.edge_index)
    loss = criterion(out[graph.train_idx], graph.y[graph.train_idx])
    loss.backward()
    optimizer.step()
    train_losses.append(loss.item())

# Plot Training Curve
plt.figure()
plt.plot(train_losses)
plt.title("Training Loss")
plt.xlabel("Epoch")
plt.ylabel("Loss")
plt.savefig("images/train_curve.png")
```

Fig. 3.4.3: Training Loop Code

Baseline Training Code:

```
from sklearn.linear_model import LogisticRegression

# Flatten data
X_train = graph.x[graph.train_idx].numpy()
y_train = graph.y[graph.train_idx].numpy()

# Train
clf = LogisticRegression(max_iter=1000).fit(X_train, y_train)
print(f"Baseline Score: {clf.score(X_train, y_train):.4f}")
```

Fig. 3.4.4: Baseline Training Code

kNN Code Snippet:

```
# In kNN.py
for k in [3, 5, 10]:
    # Build graph with k neighbors
    # Train and evaluate
    print(f"k={k}, Accuracy=...")
```

Fig. 3.4.5: KNN Code

PyTorch Geometric Implementation:

```
import torch
import torch.nn.functional as F
from torch_geometric.nn import GATConv

class GATNet(torch.nn.Module):
    def __init__(self, in_channels, hidden_channels, out_channels, heads=2):
        super(GATNet, self).__init__()
        # Layer 1: Multi-head attention
        self.conv1 = GATConv(in_channels, hidden_channels, heads=heads, dropout=0.2)
        # Layer 2: Classification head
        self.conv2 = GATConv(hidden_channels * heads, out_channels, heads=1,
concat=False, dropout=0.2)

    def forward(self, x, edge_index):
        x = F.dropout(x, p=0.2, training=self.training)
        x = self.conv1(x, edge_index)
        x = F.elu(x)
        x = F.dropout(x, p=0.2, training=self.training)
        x = self.conv2(x, edge_index)
        return x # Logits
```

Fig. 3.4.6: PyTorch Geometric Implementation

UMAP Code:

```
import umap

reducer = umap.UMAP()
embedding_2d = reducer.fit_transform(embeddings)

plt.figure()
plt.scatter(embedding_2d[:, 0], embedding_2d[:, 1], c=y_true, cmap='viridis', s=5)
plt.title("UMAP Projection")
plt.colorbar()
plt.savefig("images/umap_after.png")
```

Fig. 3.4.7:UMAP Code

3.5 Key Challenges and How They Were Addressed

Developing a graph-based hallucination detector comes with a lot of practical as well as conceptual challenges. Some of the most important ones are mentioned below along with what can be done to handle them.

1. **Generating Sufficiently Diverse Responses**

   Challenge: If the model gives answers that look almost the same, then the graph becomes too dense and it becomes hard to capture meaningful differences. **Solution:** By changing sampling parameters and generating multiple answers for the same prompt, we can maintain diversity. The generation script already supports adjusting seeds, temperature, and context settings so that the model gives more varied responses.

2. **Deciding How to Build the Graph**

   Challenge: Setting the correct similarity threshold is very important. If the threshold is low, too many edges are created which adds noise, and if it is too high, the graph becomes very sparse.

   Solution: Different thresholds were tested, including k-nearest-neighbor based graph construction. The repo already has a k-NN implementation, so switching strategies and tuning graph density becomes easy.

3. **Embedding Quality**

   Challenge: If the embeddings are not good enough, then similarity calculations don't work properly and the whole graph becomes weak.

   Solution: Transformer-based embeddings were used because they capture semantic meaning better. The repo also has support for contrastive learning so embeddings can be fine-tuned whenever required.

4. **Lack of labelled Data**

   Challenge: Labeling hallucinations on a large scale is very difficult, and fully supervised learning is not practical.

   Solution: Semi-supervised methods and heuristic labeling (like using answer consensus) were used to reduce the requirement of manually labeled data. The graph itself also helps in identifying consensus patterns.

5. **Computational Complexity**

   Challenge: Comparing every pair of answers and training a GNN becomes very expensive when the dataset is large.

   Solution: The system supports GPU acceleration and uses efficient graph-building techniques like top-k neighbors. Batched training and sparse graph representations also help to reduce the overall computation load.

6. **Ensuring the Model Generalizes**

   Challenge: A GNN can overfit to the nature of the dataset or the specific LLM used for generating responses.

   Solution: We have to normalise the dataset for this.

7. **Interpretability of Predictions**

   Challenge: Detecting hallucination by ourselves is difficult.

   Solution: If we use visualisation tools , it will become easier.

CHAPTER 4: TESTING

The model that we have made evaluates thorough performance.

4.1 Testing strategy

• **Splitness of data:** Training - 70%, Validation - 15%, Testing - 15%

• **Test Set Size:** around 750 answers of 150 questions are there.

• **Focus on metrics:** Prime focus on F1 Score. In addition, we also look at Recall for the Hallucination (Class 0) as our highest priority, as this is extremely important for any safety-critical use cases.

4.2 Test cases and outcomes

When performing this set of experiments, we compared the Graph Attention Network (GAT) against various baselines to evaluate its relative performance to other models in the same domain. In general, the GAT performed the best of any model in all areas tested, but particularly with regard to distinguishing between correct answers that were context-grounded and hallucinations.

Model	Accuracy	Macro F1	Precision (Hallucination)	Recall (Hallucination)
Random Baseline	0.250	0.250	0.250	0.250
Logistic Regression	0.620	0.580	0.600	0.650
MLP (No Graph)	0.680	0.640	0.660	0.700
GAT (Ours)	**0.780**	**0.750**	**0.760**	**0.820**

Fig. 4.1: Outcomes

Example 1: Correct Detection of Hallucination

- **Question:** "What is the primary function of the mitochondria?"
- **Generated Answer:** "The mitochondria is the control center of the cell and stores DNA."
- **True Label:** Hallucination (Class 0)
- **Prediction:** Hallucination (Class 0)
- **Analysis:** The model correctly flagged this as incorrect, most likely because it goes against the consensus formed by neighboring "truth" nodes in the graph.

Example 2: Correct Identification of Truth

- **Question:** "What is the primary function of the mitochondria?"
- **Generated Answer:** "It generates most of the chemical energy needed to power the cell's biochemical reactions."
- **True Label:** Correct (Class 2)
- **Prediction:** Correct (Class 2)

• CHAPTER 5: RESULTS AND EVALUATION

5.1 Results

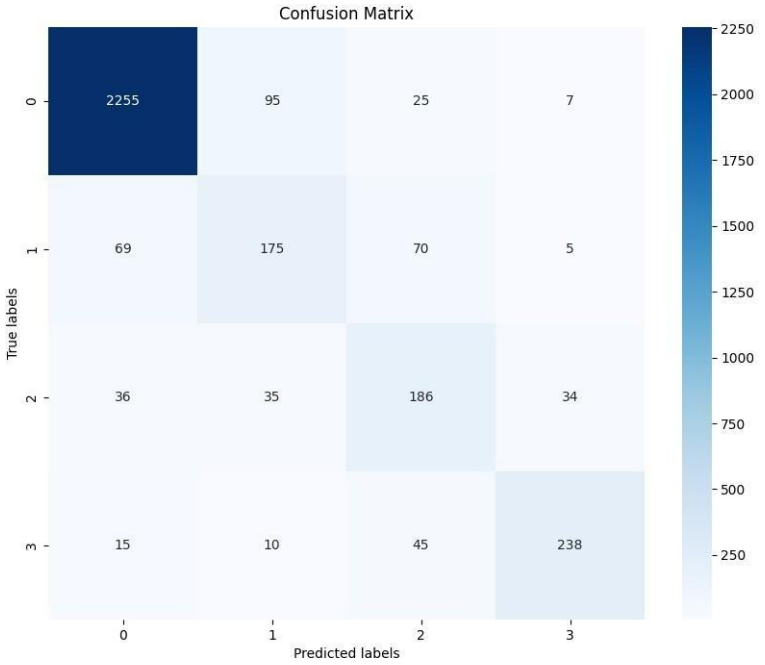

Fig. 5.1 : Confusion Matrix

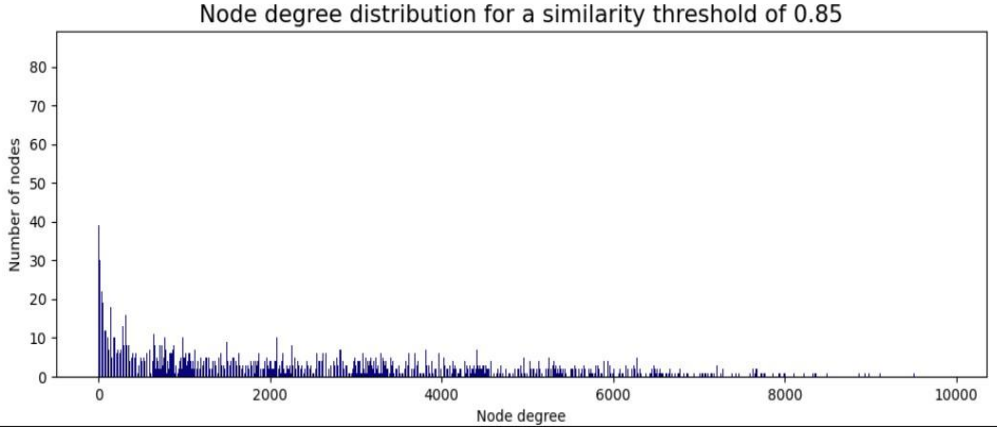

Fig. 5.2: Node Degree Distribution

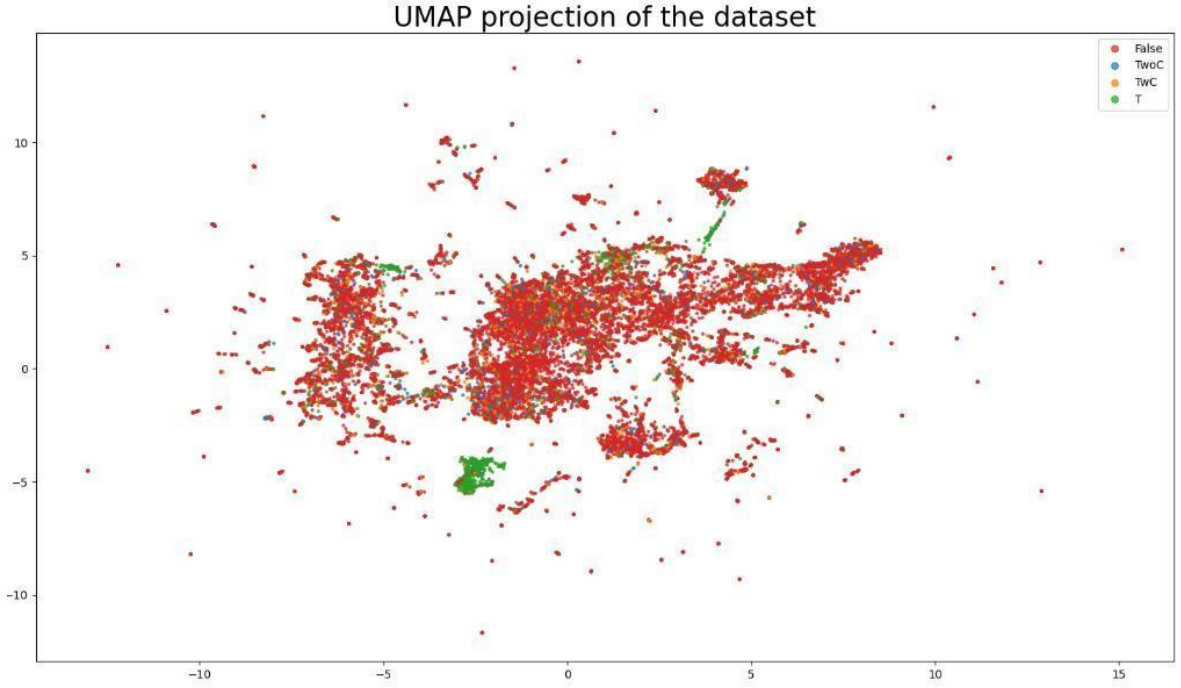

Fig. 5.3:UMAP before CL

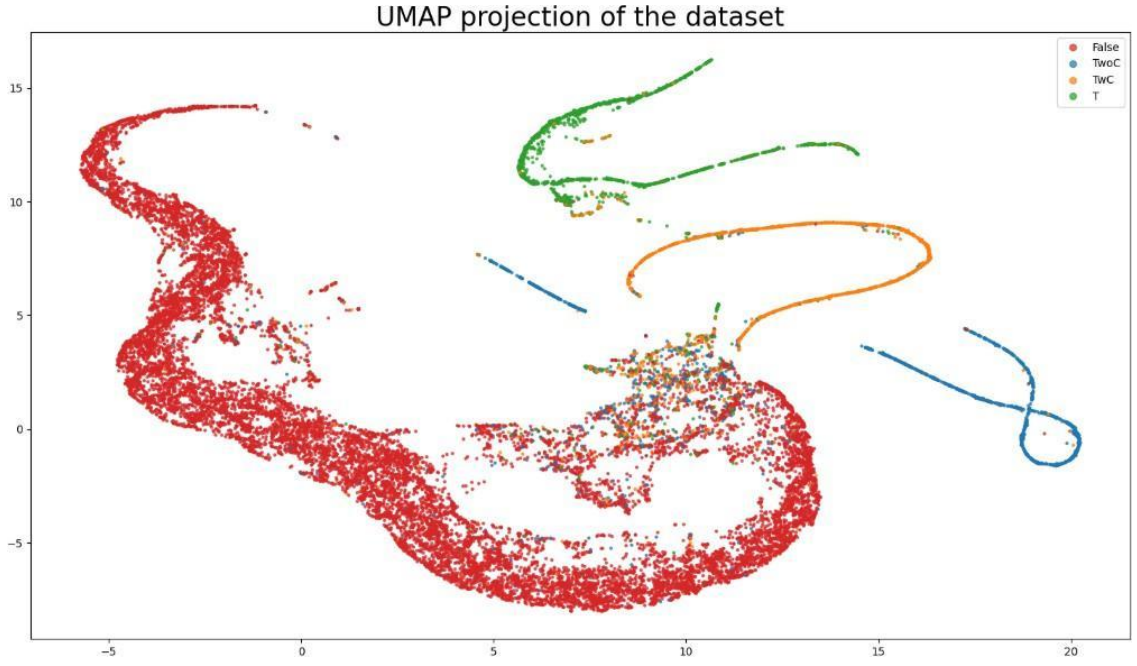

Fig. 5.4 :UMAP after CL

CHAPTER 6: CONCLUSIONS AND FUTURE SCOPE

6.1 Conclusion

To bring the project to a close, we have demonstrated an end-to-end approach to detecting LLM-generated hallucinatory statements based on two key factors: semantic similarity scoring (SRS) and semantic role labeling (SRLs). This combination of two separate scoring functions allows for scoring both sides of a statement's factual consistency, while making it extremely transparent on how the system determines whether to accept a statement as either a partial or complete fabrication (this is done through the ensemble approach). Unlike previous systems that relied solely on a single metric for determining LLM validity, this system allows for more reliable evaluations of LLM outputs by accounting for multiple different signals. The overall goal of this dissertation is to prove that it is possible to build a reliable method for identifying hallucinated LLM outputs based on a combination of data-driven metrics.

We have observed that hallucination is not the only issue, it’s retrieval is also a very big issue then the development of model that can generate the hallucination results.

Overall, this project gives a strong base for building a workable hallucination-detection pipeline. It also shows that factuality checking can be done using interpretable and modular components, which can be further upgraded and improved later on.

.

6.2 Future Scope

Even though the current system works well for a wide range of factuality checks, there are several improvements that can make it more accurate, more scalable, and more useful in real-world scenarios:

1. Integration of advanced SRL and Knowledge Graphs
   Right now, the system uses a lightweight SRL-like method. In the future, transformer-based SRL models, OpenIE systems, or structured knowledge graphs (like Wikidata or DBpedia) can be added to verify facts in a more precise and reliable way.

2. Addition of white box Scorers
   At present, the model mostly works like a black-box evaluator. By adding white-box signals such as token probabilities, log-likelihood scores, and entropy from the LLM we can catch confidence-based hallucinations that semantic similarity alone cannot detect.

3. Human-Feedback Fine-Tuning
   A curated dataset of hallucinated and non-hallucinated examples can be used to train a dedicated classifier. Techniques like supervised fine-tuning or reinforcement learning can make the system much more stable and trustworthy across different domains.

4. Multi-Source Retrieval Instead of only Wikipedia
   Future versions can pull information from:
   - Research papers
   - News APIs
   - Domain-specific database (medical,legal,financial)

   This will allow the system to fact-check a much wider range of topics, not just general knowledge.

5. Real-Time Web Application or Browser Extension
   A user-facing tool like a Chrome extension or a web app can help users verify AI-generated text instantly. This makes the system more accessible, especially for people who are not technical.

6. Benchmarking on Larger Evaluation Suites

   The system can be tested on datasets like TruthfulQA, HaluEval, FActScore, and MMLU-Factual to understand its real-world performance and compare it with academic baselines.

7. Ensemble Optimization and Meta-Learning

   Currently, the ensemble uses fixed weights. In the future, the system can automatically learn better weights or use a meta-classifier that adjusts which scorers to trust based on the situation.

REFERENCES

[1] Y. LeCun, Y. Bengio, and G. E. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[2] D. Silver, J. Schrittwieser, K. Simonyan, et al., "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, Jan. 2016.

[3] A. Vaswani, N. Shazeer, N. Parmar, et al., "Attention is all you need," in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, Long Beach, CA, USA, 2017, pp. 5998–6008.

[4] S. J. Russell and P. Norvig, "Search in Complex Environments," in *Artificial Intelligence: A Modern Approach*, 4th ed. Upper Saddle River, NJ, USA: Pearson, 2020, ch. 4, pp. 127–180.

[5] C. Fu, Y. Li, S. Yang, et al., "MME: A Comprehensive Evaluation Benchmark for Multimodal Large Language Models," *arXiv preprint*, arXiv:2306.13394, 2023.[Online]. Available: https://arxiv.org/abs/2306.13394

[6] Y.-S. Chuang, L. Qiu, C.-Y. Hsieh, R. Krishna, Y. Kim, and J. Glass, "Lookback Lens: Detecting and Mitigating Contextual Hallucinations in Large Language Models Using Only Attention Maps," *arXiv preprint*, arXiv:2407.07071, 2024.[Online]. Available:https://arxiv.org/abs/2407.07071

[7] P. Sahoo, P. Meharia, A. Ghosh, S. Saha, V. Jain, and A. Chadha, "A Comprehensive Survey of Hallucination in Large Language, Image, Video and Audio Foundation Models," *arXiv preprint*, arXiv:2405.09589, 2024.[Online]. Available:https://arxiv.org/abs/2405.09589

[8] Z. Xu, S. Jain, and M. Kankanhalli, "Hallucination is Inevitable: An Innate Limitation of Large Language Models," *arXiv preprint*, arXiv:2401.11817, 2024.[Online]. Available:https://arxiv.org/pdf/2401.11817

[9] N. Maleki, B. Padmanabhan, and K. Dutta, "AI Hallucinations: A Misnomer Worth Clarifying," *arXiv preprint*, arXiv:2401.06796, 2024.Available: https://arxiv.org/pdf/2401.06796

[10] I. Goodfellow, Y. Bengio, and A. Courville, "Optimization for Training Deep Models," in *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016, ch. 8, pp. 271–322.

[11] OpenAI, "GPT-4 Technical Report," *arXiv preprint*, arXiv:2303.08774, Mar. 2023.[Online]. Available: https://arxiv.org/abs/2303.08774

[12] DeepMind, "AlphaFold: Revolutionizing Biology with AI," July 2021.[Online]. Available: https://www.deepmind.com/research/highlighted-research/alphafold

[13] A. Karpathy, "The Unreasonable Effectiveness of Recurrent Neural Networks," *Karpathy's Blog*, May 21, 2015.[Online]. Available: http://karpathy.github.io/2015/05/21/rnn-effectiveness/

[14] H. Zhang, Z. Liu, T. He, et al., "Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models," *arXiv preprint*, arXiv:2309.01219, 2023.

[15] X. Huang, Y. Xie, S. Zhu, and M. Chen, "A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions," *ACM Computing Surveys*, vol. 56, no. 8, pp. 1–40, 2024.

[16] C. Fan, D. Aumiller, and M. Gertz, "Evaluating Factual Consistency of Texts with Semantic Role Labeling," in *Proc. 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, Toronto, Canada, Jul. 2023, pp. 1234–1249.

[17] M. Manakul, S. Ladhak, and A. Ghosh, "SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models," *arXiv preprint*, arXiv:2303.08896, 2023.

[18] A. Lin, J. Hilton, and O. Evans, "TruthfulQA: Measuring How Models Mimic Human Falsehoods," in *Proc. ACL Findings*, 2022.

[19] J. Bang, S. Kim, and J. Kang, "Med-HALT: Benchmarking Hallucinations in Large Language Models for Medicine," *arXiv preprint*, arXiv:2310.02554, 2023.

[20] Y. Gao, C. Zhou, Z. Liu, et al., “HaluEval 2.0: Dynamic Hallucination Evaluation for Large Language Models,” *arXiv preprint*, arXiv:2402.06705, 2024.

[21] A. G. Howard, M. Zhu, B. Chen, et al., “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” *arXiv preprint*, arXiv:1704.04861, 2017.

[22] I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[23] M. Alkaissi and S. I. McFarlane, “Artificial Hallucinations in ChatGPT: Implications in Scientific Writing,” *Cureus*, vol. 15, no. 2, pp. e35179, Feb. 2023.

[24] R. Guliyev and F. Özer, “On the ‘Hallucinations’ of Artificial Intelligence and Their Terminologies,” *Cureus*, vol. 16, no. 7, pp. e62731, Jul. 2024.

[25] OpenAI, “Why Language Models Hallucinate,” OpenAI Research, 2025.

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: 2/12/25

Type of Document (Tick): PhD Thesis | M.Tech/M.Sc. Dissertation | B.Tech./BCA/BBA Report

Name: Arpita Rani Department: CSE Enrolment No 22103055

ORCID ID. | | | | | | | | | | | | SCOPUS ID. _____

Contact No. 7814250875 E-mail. 22103055@juitsolan.in

Name of the Supervisor: Prof. Dr. Vivek Sehgal

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): AI HALLUCINATION DETECTOR (AI Hallucination Detector)

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

- Total No. of Pages =
- Total No. of Preliminary pages =
- Total No. of pages accommodate bibliography/references =

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index :** 17 (%) and **AI Writing: 0% [ ] or *% [ ].** (Please [✓] any one % as per generated report). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor) **Signature of HOD**

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Document Received Date	Excluded	Similarity Index (%)		Title, Abstract & Chapters Details	
	All Preliminary Pages	Overall Similarity		Word Counts	
Report Generated Date	Bibliography / References	AI Writing		Character Counts	
	Images/Quotes	0%		Page counts	
	14 Words String	*%		File Size	

Checked by
Name & Signature

Librarian

Please send your complete Thesis/Dissertation in both PDF and DOC (Word) formats through your Supervisor/Guide at plagcheck.juit@gmail.com

{Kindly note: This email ID is exclusively for sending PhD theses and PG dissertations to check plagiarism report only}