Language Model Hallucination Detection (Overleaf)

1. Overview

1.1. Problem Statement

Language models (LMs) frequently produce outputs that are syntactically convincing but factually incorrect or ungrounded in reliable sources. This phenomenon, referred to as **hallucination**, presents significant challenges in ensuring the reliability, trustworthiness, and practical adoption of AI systems in critical areas such as healthcare, education, and policy-making. The focus is on tracking AI-generated outputs, evaluating their **groundedness**, and ensuring their **faithfulness** to reliable sources to distinguish between real data and synthetic outputs.

1.2. Key Terms

- **Hallucination:** Outputs generated by a language model that are factually inaccurate or ungrounded in training data or real-world knowledge.
- **Groundedness:** The extent to which an AI's outputs can be traced back to reliable and credible sources.
- **Faithfulness:** The alignment between generated content and the source data it is derived from.
- **Synthetic Language Data:** Text created by AI models rather than written by humans.
- **Traceability:** The ability to track the origin of an AI-generated text and verify its authenticity.

2. Challenges

- **Defining and Quantifying Hallucination:**
  Lack of universally accepted definitions and metrics for hallucination leads to inconsistencies in evaluation and comparison of methods.
- **Detection and Attribution of Hallucination:**
  i. Differentiating between true and synthetic data, especially in large-scale corpora, is challenging.
  ii. Language models often blend accurate facts with hallucinations, making detection harder.
- **Evaluation Benchmarks:**
  Current benchmarks focus on synthetic tasks but fail to generalize to real-world use cases or specialized domains like healthcare or law.
- **Scalability and Real-Time Analysis:**
  Scaling hallucination detection methods to handle the output of state-of-the-art models in real time is computationally intensive.
- **Lack of Grounded Data:**
  Models trained on noisy or biased data inherit these flaws, exacerbating hallucination problems.

- **Ethical and Legal Concerns:**
  Misidentifying or misattributing content as hallucinated can have legal and reputational implications.

3. Evaluation Metrics and Benchmarks

3.1. Metrics:

- **Precision and Recall:** Measuring how accurately hallucinations are detected.
- **Faithfulness Scores:** E.g., QAGS (Question-Answering Faithfulness Score), where outputs are evaluated against a reference document.
- **Factual Consistency:** Evaluated via human annotations or automated tools like TruthfulQA or FactCC.
- **Groundedness Metrics:** Assess the extent to which generated content is anchored in reliable sources.

3.2. Benchmarks:

- **TruthfulQA:** Focused on evaluating truthfulness in language models.
- **FEVER (Fact Extraction and Verification):** A dataset designed to evaluate fact-checking models.
- **REALM:** Retrieval-Augmented Language Model benchmark that measures groundedness.
- **XSum Faithfulness Dataset:** Targets hallucinations in summarization tasks.
- **SITUATEDQA:** An open-retrieval QA dataset where systems must produce the correct answer to a question given the temporal or geographical context.

Paper Summaries (click here)

Method	Usage Context	Performance	Source Code	Commercial Use	Maintenance
TruthfulQA: Measuring How Models Mimic Human Falsehoods **2022**	A benchmark with a focus on identifying when models mimic human falsehoods and GPT-judge (LM QA). The fine-tuned models should be used as a metric for TruthfulQA only and are not expected to generalize to new questions.	~90-95% validation accuracy across all model classes but highly specific for truth evaluation (education, policy-making, and customer support systems)	Github Colab	Apache-2.0 license	Actively maintained (January 2025)
REALM **ORQA**	Combine masked language models with a differentiable retriever.	-Only explored open-domain extractive question answering	REALM GitHub ORQA GitHub	Apache-2.0 license	

Retrieval Augmented Generation for Knowledge Intensive NLP Tasks **2020**	A method designed to enhance the knowledge retrieval capabilities of LMs for tasks requiring high factual accuracy (knowledge-intensive QA and fact-based summarization). **General-purpose** fine-tuning -pre-trained **parametric memory** (BART seq2seq model) **-non-parametric memory** (Wikipedia).	+SOTA performance on Open-Domain QA tasks; grounded outputs via retrieval. -Computationally expensive and dependent of retrieval corpus quality. - https://aclanthology.org/2021.emnlp-main.586/	Hugging Face	Apache-2.0 license	Well maintained (March 2024)
Retrieval Augmentation Reduces Hallucination in Conversation **2021**	Various types of architectures with multiple components – retrievers, rankers, and encoder-decoders – with the goal of maximizing knowledgeability while retaining conversational ability.	+SOTA on two knowledge-grounded conversational tasks	Parl.ai	MIT license	
On Faithfulness and Factuality in Abstractive Summarization **2020**	Introduces new evaluation frameworks and datasets. -Pretrained models are better summarizers in terms of being faithful and factual (both ROUGE & humans).	+Textual entailment measures better correlate with faithfulness than standard metrics. -High hallucination rates in existing models -> costly manual evaluations.	Github	Creative Commons Attribution 4.0 International (CC BY 4.0).	5 years ago
SITUATEDQA: Incorporating Extra-Linguistic Contexts into QA **2021**	An open-retrieval QA dataset where systems must produce the correct answer to a question given the temporal or geographical context. A significant proportion of information seeking questions have context-dependent answers (e.g., roughly 16.5% of NQ-Open).	+Answers to the same question may change depending on the extra-linguistic contexts +Existing models struggle with producing answers that are frequently updated or from uncommon locations.	GitHub		4 years ago

# Topic I.e. Language Model Hallucination Detection

# Overview

Give problem statement.

Define key terms as needed.

# Challenges

Describe challenges to the problem. Highlight areas currently unsolved even by top approaches.

# Evaluation Metrics and Benchmarks

Describe common benchmarks and metrics used to evaluate them.

# Survey of Methods

Describe usage context and performance. Evaluation criteria: Is the method performing SOTA on current benchmarks? How fast is it? is there source code available? Is it licensed for commercial use?(mit, apache, not gpl) Does it look maintained?

Provide standalone code to call this method if applicable.

If you wrote your own use case such as a test prompt, please provide it with outputs from each method.

# Recommendations from Findings

Recommend a single method.

Provide additional suggestions such as how to improve on existing results, from fast to test to hard.