LLM Hallucination Mitigation Checklist

Type: Practical Checklist

Target Audience: AI Engineers, Product Managers, QA Teams

This checklist provides practical strategies for preventing, detecting, and managing hallucinations in Large Language Model (LLM) applications. Hallucinations are confident but incorrect outputs that can cause significant harm in production systems.

Key Principle: Assume LLMs will hallucinate. Design systems with this expectation and implement multiple layers of defense.

1. Prevention Strategies

Reduce the likelihood of hallucinations through system design and prompt engineering.

Prompt Engineering

- ☐ Use clear, specific instructions with explicit constraints
- ☐ Include examples of correct outputs (few-shot prompting)
- ☐ Instruct model to say "I don't know" when uncertain
- ☐ Request step-by-step reasoning (chain-of-thought)
- ☐ Specify output format to constrain responses
- ☐ Include domain context and relevant background information

Retrieval Augmented Generation (RAG)

- ☐ Ground responses in retrieved source documents
- ☐ Include source citations in outputs
- ☐ Limit responses to information present in retrieved context
- ☐ Implement relevance scoring for retrieved documents
- ☐ Regularly update and validate knowledge base

Model Selection and Configuration

- ☐ Choose models appropriate for the task complexity
- ☐ Use lower temperature settings for factual tasks (0.0 - 0.3)
- ☐ Consider fine-tuned models for domain-specific applications
- ☐ Test multiple models and compare hallucination rates

2. Detection Methods

Implement mechanisms to identify hallucinations before they reach end users.

Automated Detection

- ☐ Implement fact-checking against authoritative sources
- ☐ Use secondary LLM to verify primary LLM outputs
- ☐ Check for internal consistency within responses
- ☐ Validate citations and references exist
- ☐ Flag responses with low confidence scores
- ☐ Implement entity verification for names, dates, numbers

Human Review

- [ ] Require human review for high-stakes outputs
- [ ] Implement spot-checking for representative samples
- [ ] Enable user feedback mechanisms to report errors
- [ ] Train reviewers on common hallucination patterns

3. Common Hallucination Types

Understand the different categories of hallucinations to better detect and prevent them.

Type	Description	Detection Strategy
Factual Errors	Incorrect facts, dates, statistics, or claims	Cross-reference with authoritative databases
Fabricated Citations	Made-up references, papers, or quotes	Verify citations exist in academic/news databases
Entity Confusion	Mixing up people, places, or organizations	Entity resolution against knowledge graphs
Temporal Errors	Incorrect dates or anachronistic information	Timeline validation; recency checks
Logical Inconsistency	Self-contradicting statements within response	Consistency checking across response
Overconfident Uncertainty	Presenting guesses as definitive facts	Calibration testing; uncertainty prompting

4. Response Protocols

Define clear procedures for when hallucinations are detected.

Immediate Response

- [ ] Block or flag output before delivery to user
- [ ] Log the hallucination for analysis
- [ ] Regenerate response with modified prompt
- [ ] Escalate to human review if regeneration fails

User Communication

- [ ] Display confidence indicators to users
- [ ] Provide disclaimers for AI-generated content
- [ ] Enable easy reporting of suspected errors
- [ ] Offer source verification where applicable

5. Ongoing Monitoring

- [ ] Track hallucination rates by category and use case
- [ ] Monitor user feedback and error reports
- [ ] Conduct regular audits with human evaluators

- ☐ Benchmark against new model versions
- ☐ Update detection rules based on observed patterns
- ☐ Report metrics to stakeholders regularly

System Name: _____

Reviewed By: _____ **Date:** _____