Comprehensive Notes: Minimizing AI Hallucinations - Practical Techniques

I. FOUNDATIONAL PRINCIPLES

Core Understanding

- **Root Cause**: Hallucinations occur because models fill information gaps
- **Solution Strategy**: Force grounding, reduce randomness, add verification layers
- **Implementation**: System prompts + user prompts + RAG workflows + parameter tuning + testing

Six Core Principles (Critical Foundation)

1. **Ground First**
    - Provide exact documents, facts, or API results
    - Instruct model to use ONLY provided sources
    - No general knowledge fallback allowed
2. **Demand Evidence**
    - Require numbered citations for every factual claim
    - Include direct quotes from sources
    - Provide URLs where applicable
3. **Explicit Uncertainty Handling**
    - Force "I don't know" responses when unsupported
    - Mark unsupported claims clearly
    - Prefer incomplete answers over invented facts
4. **Reduce Generation Randomness**
    - Use deterministic settings (low temperature)
    - Consistent parameter configuration
    - Predictable output patterns
5. **Decompose Complex Queries**
    - Break down complex questions into smaller parts
    - Reduce scope for invention
    - Handle each sub-question individually
6. **Test and Verify**
    - Implement automatic verification checks
    - Conduct regular human spot checks

- Maintain quality assurance pipeline

II. SYSTEM CONFIGURATION

Essential System Message Template

```
SYSTEM:
You are a fact-focused assistant. Only use information provided in the "context" section or explicitly cited high-quality sources. For each factual statement, include a numeric citation [1], [2], ... that points to a source listed in the Evidence block. If any statement cannot be supported by those sources, write "UNSUPPORTED" instead of inventing facts. At the end, provide:
- Evidence block: numbered sources with exact quoted snippets used.
- Confidence score (0-100%) for the overall answer and note which claims are uncertain.
Answer concisely and in bullet form when possible.
```

Model Parameters (API/Playground Settings)

Parameter	Recommended Value	Purpose
temperature	0.0 - 0.2	Minimize randomness; 0 = maximum determinism
top_p	0.8 - 1.0	Fine with low temperature; 1.0 acceptable
max_tokens	Keep tight	Shorter answers = less invention opportunity
n	1	Single best answer only
stop sequences	Use appropriately	Prevent runaway text generation
streaming	OK	But validate final content

Key Formula: Lower temperature + explicit instructions = significant hallucination reduction

III. PROMPT PATTERNS & TEMPLATES

A. Fact Q&A (Single-Shot, Grounded)

```
USER:
Context: [PASTE retrieved documents - label as Source 1, Source 2, ...]
Question: <your question>

INSTRUCTIONS:
- Use ONLY the Context above.
- Produce answer ≤200 words.
- For every factual sentence include numeric citation [1], [2].
- For each citation include exact quoted sentence in Evidence block.
- If no source supports claim, mark "UNSUPPORTED".
- End with "Confidence: X%".
```

B. RAG Pipeline Synthesis (Recommended for Large Knowledge)

```
USER:
Step 1: Retrieve top 5 documents for query (done externally).
Step 2: Provided Documents: [Source1, Source2, ...]
Task: Synthesize short answer (≤150 words) using only those documents.
Output format:
1) Answer (with inline citations [1],[2] for each claim).
2) Evidence block: for each citation include exact quoted snippet and URL.
3) Unsupported claims: list if any.
4) Confidence: X%.
```

C. Summarization with Strict Provenance

```
USER:
Context: [Article text]
Task: Produce 5-bullet summary. Each bullet must include:
(a) One-line summary sentence
(b) Exact supporting quote in parentheses with source citation
No extra claims allowed.
```

D. Code Generation + Verification

```
USER:
Task: Write function X. Also:
- Include at least 3 unit tests demonstrating expected behavior.
- After code, list exact sources or docs used (if any).
- If any behavior based on assumptions, mark UNSUPPORTED and list how to confirm.
```

IV. RETRIEVAL ENGINEERING STRATEGY

Document Retrieval Process

1. **Initial Retrieval**
   - Use semantic search with embeddings
   - Retrieve k = 5-10 documents (depends on length)
   - Consider query complexity
2. **Document Processing**
   - Chunk long documents to <1,500 tokens
   - Maintain 10-20% semantic overlap between chunks
   - Preserve context across chunk boundaries

3. **Re-ranking Strategy**
    - Prioritize exact keyword matches
    - Consider document recency when applicable
    - Use relevance scoring
4. **Prompt Integration**
    - Label chunks clearly (Source 1, Source 2, etc.)
    - Include metadata: title, date, URL
    - Maintain full snippet availability for Evidence blocks

Quality Assurance Requirements

- Always include full snippet for any factual claim
- Preserve source traceability
- Maintain metadata consistency

V. VALIDATION & VERIFICATION FRAMEWORK

Core Verification Steps (Required)

1. **Claim Extraction**
    - Request numbered list of discrete factual claims
    - Ensure each claim is independently verifiable
    - Maintain claim granularity
2. **Source Matching**
    - Require exact quoted snippet for each claim
    - Provide source ID for traceability
    - Verify quote accuracy
3. **Unsupported Flagging**
    - Mark claims without matching quotes as UNSUPPORTED
    - Prefer gaps over invention
    - Maintain integrity standards
4. **Cross-Checking**
    - Run counter-prompts: "List 3 ways this answer could be wrong"
    - Inspect for potential errors
    - Challenge answer validity
5. **Automated Testing**
    - Generate and run unit tests for technical answers

   - Implement CI pipeline integration
   - Maintain test coverage

6. **Human Quality Control**
   - Randomly sample 5-10% of answers
   - Verify source accuracy
   - Maintain quality metrics

Verification Prompt Template

```
USER:
Now produce:
- Numbered list of all factual claims you made.
- For each claim, paste exact supporting quote and source ID used. If none, write UNSUPPORTED.
- Short justification (1 sentence) of confidence for each claim.
```

VI. ADVANCED TECHNIQUES

Few-Shot Learning Implementation

- **Purpose**: Model learns pattern through examples
- **Structure**: Provide one good example (with evidence) and one bad example (invented claim)

Example Template:

```
Good:
Q: Is X true?
A: Yes [1]. Evidence: "exact quote..." (Source 1)

Bad:
Q: Is Y true?
A: Y is true. (No evidence) — This is INVALID.
```

High-Stakes Content Handling (Medical/Legal/Financial)

- **Never allow general knowledge responses**
- **Require primary sources**: guidelines, laws, peer-reviewed studies
- **Enhanced system prompt addition**:

  ```
  "For any clinical or legal recommendation, include full citations to primary sources and add 'Not medical/legal advice — consult a professional'."
  ```

- **Force uncertainty acknowledgment** when sources insufficient

VII. DEBUGGING METHODOLOGY

Practical Debugging Checklist

1. **Temperature Test**
   - Re-run with temp=0
   - If claim disappears → likely invented by randomness
2. **Quote Verification**
   - Require direct quotes for all claims
   - Missing quotes = model guessing
3. **Scope Narrowing**
   - Rephrase to simple, fact-checkable items
   - Reduce complexity systematically
4. **Negative Examples**
   - Show what invention looks like
   - Train model to recognize bad patterns
5. **Audit Trail**
   - Log model outputs and sources
   - Calculate: support_rate = supported_claims / total_claims
   - **Target**: 0.95+ for production factual endpoints

VIII. EVALUATION METRICS

Key Performance Indicators

1. **Support Rate**
   - Formula: (# claims with supporting quote) / (total claims)
   - Target: >95% for production systems
2. **Precision**
   - Formula: (# correct supported claims) / (total supported claims)
   - Measures accuracy of supported statements
3. **Recall**
   - Formula: (# ground-truth claims found) / (total ground-truth claims)
   - Measures completeness of coverage
4. **Uncertainty Rate**
   - Formula: (# UNSUPPORTED or low confidence claims) / (total claims)
   - Measures appropriate uncertainty acknowledgment

Implementation

- Automate metrics against validation set
- Regular performance monitoring
- Continuous improvement tracking

IX. READY-TO-USE TEMPLATES

A. Evidence-First Answer Format

```
Answer (≤150 words). Inline citations [1][2].
Evidence:
[1] Title — URL
Quote: "..."
[2] Title — URL
Quote: "..."
Unsupported claims: [...]
Confidence: 85%
```

B. RAG Verification Template

```
Use only Sources 1–5. For each factual sentence, provide:
(A) The sentence
(B) Supporting source ID
(C) Exact quoted snippet
If none, write UNSUPPORTED.
```

X. IMPLEMENTATION EXPECTATIONS & LIMITATIONS

Realistic Outcomes

- **Dramatic reduction** in hallucinations (not 100% elimination)
- **Models synthesize by design** - grounding + verification is primary defense
- **Production requirements**: Automated verification essential
- **Human oversight**: Still required for high-risk outputs

Success Formula

RAG System + System Prompt + Temperature=0 + Forced Quotes = Majority of Confidently False Statements Eliminated

Best Practices Summary

1. Implement systematic approach across all components

2. Maintain consistent verification standards
3. Regular quality audits and metric monitoring
4. Continuous refinement based on performance data
5. Human oversight for critical applications

Note: These techniques are battle-tested and practical for immediate implementation. Success depends on consistent application across all system components and regular quality monitoring.