Anthropic
CCAR-F
Claude Certified Architect - Foundations (CCAR-F)

For More Information – Visit link below:
https://www.examsempire.com/

Product Version

1. **Up to Date products, reliable and verified.**
2. **Questions and Answers in PDF Format.**

Latest Version: 6.1

Question: 1

You are building a structured data extraction system using Claude. The system extracts information from unstructured documents, validates the output using JavaScript Object Notation (JSON) schemas, and maintains high accuracy. It must handle edge cases gracefully and integrate with downstream systems.
Testing reveals that when source documents are missing certain specifications, the model fabricates plausible-sounding values to satisfy your schema's required fields. For example, a document mentioning only dimensions receives a fabricated "weight: 2.3 kg" in the extraction output.
What schema design change most effectively addresses this hallucination behavior?

A. Add explicit instructions to the prompt stating "only extract information explicitly stated in the document; use placeholder text for missing values."
B. Change fields that may not exist in source documents from required to optional, allowing the model to omit them.
C. Add a "confidence" field alongside each specification where the model self-reports its certainty, then filter out low-confidence extractions.
D. Implement semantic validation that verifies each extracted value appears in or can be inferred from the source document text.

Answer: B

Explanation:
The schema is creating a structural incentive for fabrication. When a field is declared required, the output must contain a value even when the source document contains no corresponding evidence. Structured Outputs can guarantee that Claude's response conforms to a JSON Schema, but schema conformance does not establish that every generated value is factually supported. Anthropic's documentation shows that the required array determines which properties must be present; therefore, source-dependent properties that may legitimately be absent should not be included as required fields. (https://platform.claude.com/docs/en/build-with-claude/structured-outputs)
Option B corrects the problem at the contract level. Claude can omit the unavailable property rather than inventing content merely to produce valid JSON. A nullable representation could also be used when downstream systems require a stable key set, but forcing an unsupported non-null value is architecturally unsound.
Option A still requires placeholder generation and does not resolve the mismatch between the schema and available evidence. Option C relies on model-generated confidence, which is not a substitute for grounding. Option D is a useful secondary control, but it does not constitute the requested schema-design change. Anthropic recommends allowing uncertainty and requiring factual claims to be grounded in the provided material.

(https://docs.anthropic.com/en/docs/test-and-evaluate/strengthen-guardrails/reduce-hallucinations)
Official references/topics: Structured Outputs—JSON Schema design; Reduce Hallucinations—allowing uncertainty and grounding claims.

Question: 2

You are building a structured data extraction system using Claude. The system extracts information from unstructured documents, validates the output using JavaScript Object Notation (JSON) schemas, and maintains high accuracy. It must handle edge cases gracefully and integrate with downstream systems.
Your extraction pipeline processes restaurant menus and must output structured JSON with fields for item names, descriptions, prices, and dietary tags. Some menus use inconsistent formatting—prices as "$12" vs "12.00", dietary info as icons vs text.
What's the most reliable approach?

A. Use separate extraction calls for each field to ensure consistent handling of each type.
B. Define a strict output schema and include format normalization rules in your prompt.
C. Request multiple extraction attempts per document and select the most common format.
D. Extract data as-is and normalize formats in post-processing code after Claude returns.

Answer: D

Explanation:
The most reliable architecture separates semantic interpretation from deterministic normalization. Claude is well suited to identifying that "$12" and "12.00" represent prices, or that a leaf icon represents a dietary classification. However, canonical conversion—removing currency symbols, converting values to decimal types, mapping icons to controlled labels, and enforcing locale-specific rules—is more predictably performed in application code.
Structured Outputs guarantee that Claude returns valid JSON matching the supplied schema, but that guarantee concerns structural conformance. It does not by itself guarantee that every semantically equivalent source representation will be normalized identically. Anthropic's evaluation guidance identifies code-based checks as the fastest, most reliable, and most scalable mechanism for rule-based validation. (https://platform.claude.com/docs/en/build-with-claude/structured-outputs)
Option D therefore minimizes model responsibility: Claude extracts the evidence as represented, and deterministic post-processing converts it into the canonical downstream format. This also makes normalization rules independently testable, version-controlled, and auditable.
Option A increases latency and cost without solving normalization. Option B improves output structure, but prompt-based normalization can still vary across ambiguous formats. Option C introduces unnecessary stochasticity and majority-vote logic where explicit parsing rules are available. The downstream contract should remain stable, but format conversion should be implemented using deterministic transformations rather than repeated probabilistic inference.

Official references/topics: Structured Outputs—schema compliance; Evaluation Design—code-based validation; Reliable extraction pipelines.

Question: 3

You are building a structured data extraction system using Claude. The system extracts information from unstructured documents, validates the output using JavaScript Object Notation (JSON) schemas, and maintains high accuracy. It must handle edge cases gracefully and integrate with downstream systems.
Your system has been operating with 100% human review for 3 months. Analysis shows that extractions with model confidence ≥90% have 97% accuracy overall. To reduce reviewer workload, you plan to automate high-confidence extractions.
Before deploying, what validation step is most critical?

A. Analyze accuracy by document type and field to verify high-confidence extractions perform consistently across all segments, not just in aggregate.
B. Compare accuracy at different confidence thresholds (85%, 90%, 95%) to find the optimal cutoff that maximizes automation while minimizing errors.
C. Verify that 97% accuracy meets requirements for all downstream systems that consume the extracted data.
D. Run a two-week pilot routing 25% of high-confidence extractions directly to downstream systems and monitor error reports.

Answer: A

Explanation:
An aggregate accuracy value can conceal severe performance disparities. A system may achieve 97% overall accuracy while performing poorly on a low-volume document type, a critical financial field, or a specific edge case. Automating outputs solely from the aggregate figure could therefore expose downstream systems to concentrated, high-impact errors.
Anthropic's evaluation guidance states that evaluations should be task-specific, reflect the real-world task distribution, and explicitly include edge cases. It also emphasizes multidimensional success criteria rather than reliance on a single global metric.
(https://docs.anthropic.com/en/docs/build-with-claude/develop-tests) Option A applies those principles by stratifying performance according to document type and field. This reveals whether confidence is calibrated consistently and whether the proposed automation threshold remains safe for every operationally significant segment.
Option B is useful only after segment-level performance has been understood. Selecting a global threshold cannot correct a subgroup where confidence is systematically overstated. Option C is necessary governance work, but it treats the overall 97% result as though errors were uniformly distributed. Option D places unvalidated outputs into downstream systems and depends on passive error reporting, which may fail to detect silent corruption.
The correct deployment gate is therefore segmented validation, followed by threshold selection, downstream acceptance criteria, and a controlled pilot.

Visit us at: https://www.examsempire.com/ccar-f

Official references/topics: Define Success Criteria; Task-Specific Evaluations; Edge-Case Coverage; Reliability Segmentation.

Question: 4

You are building a structured data extraction system using Claude. The system extracts information from unstructured documents, validates the output using JavaScript Object Notation (JSON) schemas, and maintains high accuracy. It must handle edge cases gracefully and integrate with downstream systems.
The system routes documents with extraction confidence below 85% to human review. A quarterly audit reveals that 12% of high-confidence extractions (≥85%) also contain errors—cases where the model finds plausible-but-incorrect values. Error sources vary: comparison tables showing competitor specs, appendices referencing different product variants, and ambiguous phrasing the model misinterprets. You need a sustainable strategy to catch these high-confidence errors and measure whether improvements reduce the error rate over time.
What approach is most effective?

A. Add a verification pass that re-extracts from each high-confidence document, flagging cases where the two extraction attempts produce different results.
B. Implement heuristic rules that flag documents containing comparison tables or appendices for review regardless of confidence score.
C. Lower the confidence threshold from 85% to 70%, routing a larger volume of extractions to human review.
D. Implement stratified random sampling reviewing a fixed percentage of high-confidence extractions weekly, enabling error rate measurement and novel pattern detection.

Answer: D

Explanation:
Stratified random sampling provides both an ongoing quality-control mechanism and an unbiased measurement framework. By reviewing a fixed proportion of high-confidence outputs across meaningful strata—such as document type, field, source format, and business risk—the organization can estimate the residual error rate, compare performance across releases, and discover failure patterns that were not anticipated when existing rules were designed.
Anthropic recommends measurable success criteria and evaluations that mirror the real-world task distribution, including edge cases. Evaluation volume and repeatability are important because improvements must be demonstrated empirically rather than inferred from isolated examples. (https://docs.anthropic.com/en/docs/build-with-claude/develop-tests) A weekly sampling program creates a stable benchmark and allows confidence intervals, trend analysis, regression detection, and error-taxonomy updates.
Option A detects only disagreements between stochastic extractions. Two attempts may produce the same plausible but incorrect answer, so agreement is not equivalent to factual correctness. Option B addresses known patterns but will miss new failure modes and may generate excessive false positives. Option C actually lowers the review threshold in the wrong direction: documents between 70% and 85% would be treated as automated rather than

reviewed if the routing rule remains “below threshold,” and changing thresholds alone does not measure high-confidence error prevalence.
Official references/topics: Evaluation Design; Representative Test Distributions; Continuous Reliability Measurement; Human-in-the-Loop Sampling.

Question: 5

You are building a structured data extraction system using Claude. The system extracts information from unstructured documents, validates the output using JavaScript Object Notation (JSON) schemas, and maintains high accuracy. It must handle edge cases gracefully and integrate with downstream systems.
Your system extracts event metadata (date, location, organizer, attendee_count) from news articles using a JSON schema with all nullable fields. During evaluation, you observe the model frequently generates plausible but incorrect values for fields not mentioned in the article—for example, outputting “500” for attendee_count when the source contains no attendance information.
What’s the most effective way to reduce these false extractions?

A. Upgrade to a more capable model tier with improved instruction-following to reduce hallucination tendencies.
B. Make all schema fields required (non-nullable) with strict validation rules to ensure the model only outputs verifiable data.
C. Add prompt instructions to return null for any field where information is not directly stated in the source.
D. Add a post-processing step using a second LLM call to verify each extracted value exists in the source document.

Answer: C

Explanation:
The schema already supports the correct representation of missing evidence: null. The remaining defect is behavioral. Claude must be explicitly instructed that absence of information is a valid outcome and that values may be populated only when directly supported by the supplied article.
Anthropic’s hallucination-reduction guidance recommends explicitly permitting Claude to express uncertainty, grounding outputs in the source, and retracting claims that lack supporting evidence. It also recommends restricting the model to the provided documents rather than allowing unsupported external knowledge. (https://docs.anthropic.com/en/docs/test-and-evaluate/strengthen-guardrails/reduce-hallucinations) Option C translates those controls directly into the extraction contract: when no source evidence exists, the model returns null. Option A may improve baseline performance but does not remove the architectural ambiguity that encourages completion of missing fields. Option B is counterproductive because non-nullable required fields force the model to provide values even when the article contains none. Structured validation would confirm that the response is syntactically valid while accepting

fabricated content. Option D adds cost, latency, and another probabilistic model decision; a second model can repeat or endorse the original hallucination.
A robust implementation should supplement the instruction with source spans or quotations for populated fields and automated checks where feasible. Nevertheless, among the options, explicit null behavior is the direct and most effective correction.
Official references/topics: Reduce Hallucinations; External-Knowledge Restriction; Nullable JSON Schema Fields; Grounded Extraction.

Visit us at: https://www.examsempire.com/ccar-f

Thank You for Trying Our Product

Special 16 USD Discount Coupon: NSZUBG3X

Email: support@examsempire.com

Check our Customer Testimonials and ratings available on every product page.

Visit our website.

https://examsempire.com/