INTERACTION WITH LARGE SCALE MODELS

LIACD/1

University of Beira Interior,
Department of Informatics

Hugo Pedro Proença,
hugomcp@di.ubi.pt, 2024/2025

INTERACTION WITH LARGE SCALE MODELS

[05]

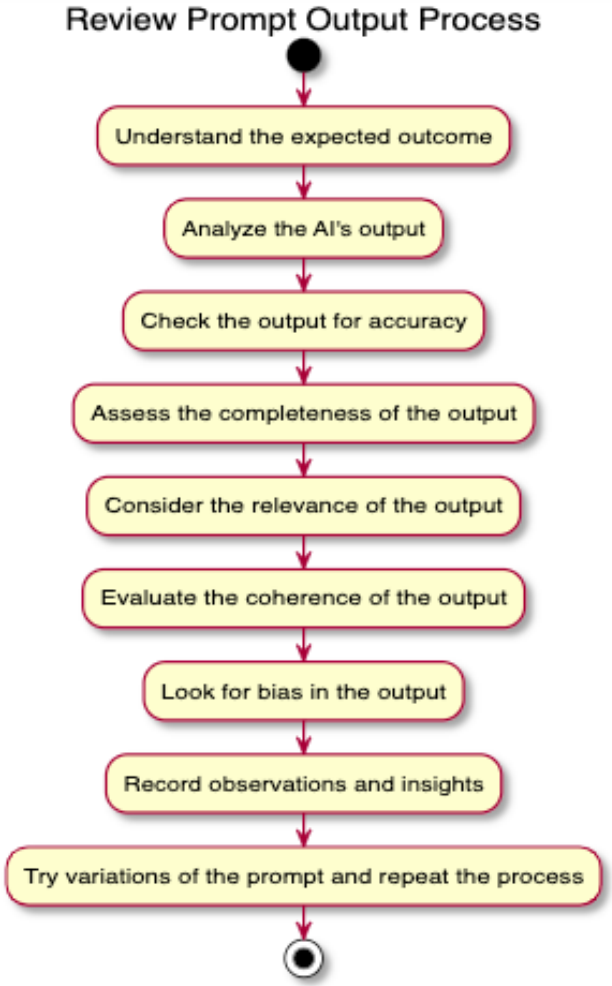
- Key metrics for prompt evaluation: Perplexity, accuracy, coherence, etc.

- Techniques for refining prompts through iterative testing.

- Tools for automated evaluation and performance tracking.

Prompt Evaluation

- Ensuring High-Quality, Reliable Model Responses
  - Evaluating how well an LLM adheres to expected outputs
  - Avoiding generic or incorrect responses in sensitive applications
- Optimizing Computational Efficiency
  - Well-structured prompts reduce unnecessary token generation
  - Helps control API costs and response latency
- Reducing Bias and Hallucinations
  - Ensuring factual accuracy and reducing misinformation
  - Identifying biases in generated text and mitigating them through prompt adjustments
- Essential for Production-Ready Applications
  - Key to ensuring LLMs meet business and academic requirements
  - Example use cases: Customer service bots, code generation, legal document analysis
  - Continuous evaluation for evolving real-world scenarios



Source: https://tonylixu.medium.com/prompt-engineering-evaluating-and-refining-your-prompt-6baf455ef451

Prompt Evaluation: Sensitive Applications

- **Medical Diagnosis Assistance**
  - A hospital integrates an LLM-based chatbot to assist doctors with preliminary diagnoses based on patient symptoms.
  - **Risk of Generic/Incorrect Responses:** If the model provides a too vague advice (*"You might have a cold"*) or incorrect suggestions (*"You have pneumonia"* when it's actually asthma), it could lead to misdiagnosis.
  - **Evaluation Strategy:**
    - Use **accuracy-based metrics** (precision, recall) to assess response correctness.
    - Implement a **fact-checking mechanism** against medical databases.
    - Ensure responses include **certainty levels and reference medical sources**.
      - Cross-check with human review/expertise
    - Compare the results against SOTA models specific to the domain of interest
      - Typically, these models might achieve comparable (or higher) performance than the foundational model.

Biased Prompts/Responses Example

Ensuring Unbiased Responses in Political or Social Topics

LLMs may generate politically biased or opinionated responses if the prompt is vague.

Weak Prompt:

"What is the best political system in the world?"

This can lead to **subjective, biased responses** based on model pretraining data.

Improved Prompt:

"Provide an objective comparison of different political systems (e.g., democracy, autocracy, monarchy) based on key factors such as governance efficiency, economic impact, and civil liberties. Use data from reputable sources, and do not express personal opinions."

- **Encourages neutrality** by requesting an objective comparison.
- **Minimizes bias** by requiring key evaluation factors.
- **Reduces hallucinations** by instructing the model to rely on **reputable sources**.

Preventing Hallucinations

LLMs may generate plausible but incorrect scientific claims when asked open-ended questions.

Weak Prompt:

"Explain the latest breakthrough in quantum computing."

The model might **hallucinate** non-existent breakthroughs or outdated research.

Improved Prompt:

"Summarize the latest developments in quantum computing based on peer-reviewed research published in the last two years. If you cannot verify the information, explicitly state that the data is unavailable."

- **Reduces hallucination risk** by demanding **peer-reviewed sources**.
- **Adds verification** by instructing the model to acknowledge missing data.
- **Promotes factual accuracy** by limiting the time frame to the **last two years**.

Qualitative/Quantitative Evaluation

	Quantitative Evaluation	Qualitative Evaluation
Definition	Uses **measurable metrics** to evaluate LLM responses.	Relies on **human judgment** for subjective aspects of response quality.
Goal	Ensure **accuracy, fluency, and relevance** in a structured way.	Assess **coherence, usefulness, creativity, and reasoning quality**.
Data Type	Numerical scores, statistical analysis.	Human feedback, expert assessment, contextual interpretation.
Scalability	Highly scalable, automated evaluation.	Harder to scale, requires domain experts.
Use Cases	Machine translation, factual QA, chatbot accuracy.	Open-ended text generation, summarization, argument quality.

Quantitative Evaluation Metrics

- **Used for structured, factual, and measurable prompt evaluations**
  - **Perplexity (PPL):** Measures how well the model predicts text (lower is better).
  - **Accuracy / Exact Match (EM):** Percentage of responses that fully match the expected output.
  - **F1 Score:** Measures precision and recall in NLP tasks.
  - **BLEU Score (MT, Text Generation):** Measures n-gram overlap between generated and reference text.
  - **ROUGE Score (Summarization):** Evaluates recall-based overlap of n-grams.
  - **Diversity Metrics (Self-BLEU, Distinct-N):** Checks response variability.
  - **Edit Distance / Levenshtein Distance:** Measures the number of changes needed to match a reference.

For most of these metrics, the “ground-truth” is the most sensitive piece of information to obtain.

Requires human-intensive labor.

Qualitative Evaluation Criteria

Used for open-ended, nuanced, and contextual assessments)

- **Coherence:** Logical consistency and fluency of the response.
- **Relevance:** Alignment with the intended prompt objective.
- **Factual Accuracy:** Whether the response contains verified, truthful information.
- **Creativity & Depth:** Especially for tasks like storytelling or brainstorming.
- **Bias & Fairness:** Whether the response is neutral and free from harmful biases.
- **Human-Likeness:** Does it sound natural, engaging, and contextually appropriate?

Quantitative Evaluation: Example 1

- Consider that we are evaluating two different prompts for generating summaries of news articles using a language model. You want to determine which prompt produces more **predictable, well-structured** outputs by measuring **Perplexity (PPL)**.
- **Definition of Perplexity**
  - Measures how well a model predicts the next token in a sequence. Lower perplexity means the model is **more confident** in its word choices, leading to **more coherent and fluent** outputs.

$$PPL = e^{\frac{-1}{N}\sum_{i=1}^{N}\log_e P(\omega_i)}$$

where $\omega_i \epsilon \{1..n\}$ is a sequence of N words (tokens)

- **Lower values** → The output is more predictable and fluent
- Higher values → The model is uncertain, leading to disjointed or incoherent responses

Quantitative Evaluation: Ex 1

- Suppose we have a language model and we want to measure the perplexity of one prompt:
- Suppose the model generates the response: "*AI is revolutionizing healthcare by improving diagnostics and treatment options*." and assigns the following word probabilities:

Word	Probability $P(w_i)$
AI	0.15
is	0.20
revolutionizing	0.10
healthcare	0.25
by	0.30
improving	0.18
diagnostics	0.12
and	0.40
treatment	0.14
options.	0.10

```
response = openai.ChatCompletion.create(
    model="gpt-4",
    messages=[{"role": "user", "content": "Explain AI's impact on
healthcare."}],
    max_tokens=50,
    logprobs=True  # Request log probabilities
)

# Extract token probabilities
tokens = response['choices'][0]['logprobs']['tokens']
log_probs = response['choices'][0]['logprobs']['token_logprobs']
```

Quantitative Evaluation: Example 1

- **Step 1: Define the Prompts**
  - **Prompt A (Unclear Instructions)**

    "Summarize the following article."
  - **Prompt B (Well-Defined Instructions with Constraints)**

    *"Summarize the following article in **two sentences**, maintaining key facts while ensuring clarity. Use formal language."*
- **Step 2: Generate responses for multiple articles using both prompts and measure PPL**

Article	Prompt A Output	PPL (Prompt A)	Prompt B Output	PPL (Prompt B)
News 1	"The economy is changing. More details inside."	**85.3**	"The economy is slowing due to inflation, as reported by financial experts."	**47.6**
News 2	"A big storm hit. Weather was bad."	**91.2**	"A severe storm impacted the city, causing damage and power outages."	**50.8**

Quantitative Evaluation: Example 2

- Consider now the scenario of evaluating Prompts for a Question-Answering (QA) System
- Suppose that a team is developing a customer support chatbot and wants to evaluate how well different prompts affect the accuracy of the model's responses.
- They use the **F1 score** to measure how closely the chatbot's answers match the expected (ground truth) responses.
- **Step 1: Define the Prompts**
  - **Prompt A (Vague Prompt)**

    "Answer the customer's question."
  - **Prompt B (Well-Defined Prompt with Constraints)**

    *"Answer the customer's question **concisely and accurately**, using no more than **three sentences**, and **include relevant product details if applicable**."*

Quantitative Evaluation: Example 2

The chatbot is then tested on a **dataset of 100 customer queries**, where each query has a **ground truth**.

The F1 score is calculated by comparing the chatbot's generated answers with these ground truth responses.

Example Responses for a Customer Query

Customer Question: *"Does the new smartphone model have wireless charging?"*

- **Ground Truth Answer:** *"Yes, the new model supports wireless charging with Qi-certified chargers."*
- **Model Output (Prompt A):** *"Yes, it does."*
  - 🔴 **Weakness:** Too short, lacks product details.
- **Model Output (Prompt B):** *"Yes, the new model supports wireless charging using Qi-certified chargers for fast charging."*
  - ✅ **Strength:** More informative, matches ground truth better.

Quantitative Evaluation: Example 2

The F1 score is given by:

$$F1 = \frac{2 \times Precision \times Recall}{\text{Precision} + \text{Recall}}$$

where **Precision** = Correct words in model response / Total words in model response and **Recall** = Correct words in model response / Total words in ground truth

Prompt	Precision	Recall	F1 Score
Prompt A	0.60	0.50	**0.54**
Prompt B	0.85	0.90	**0.87**

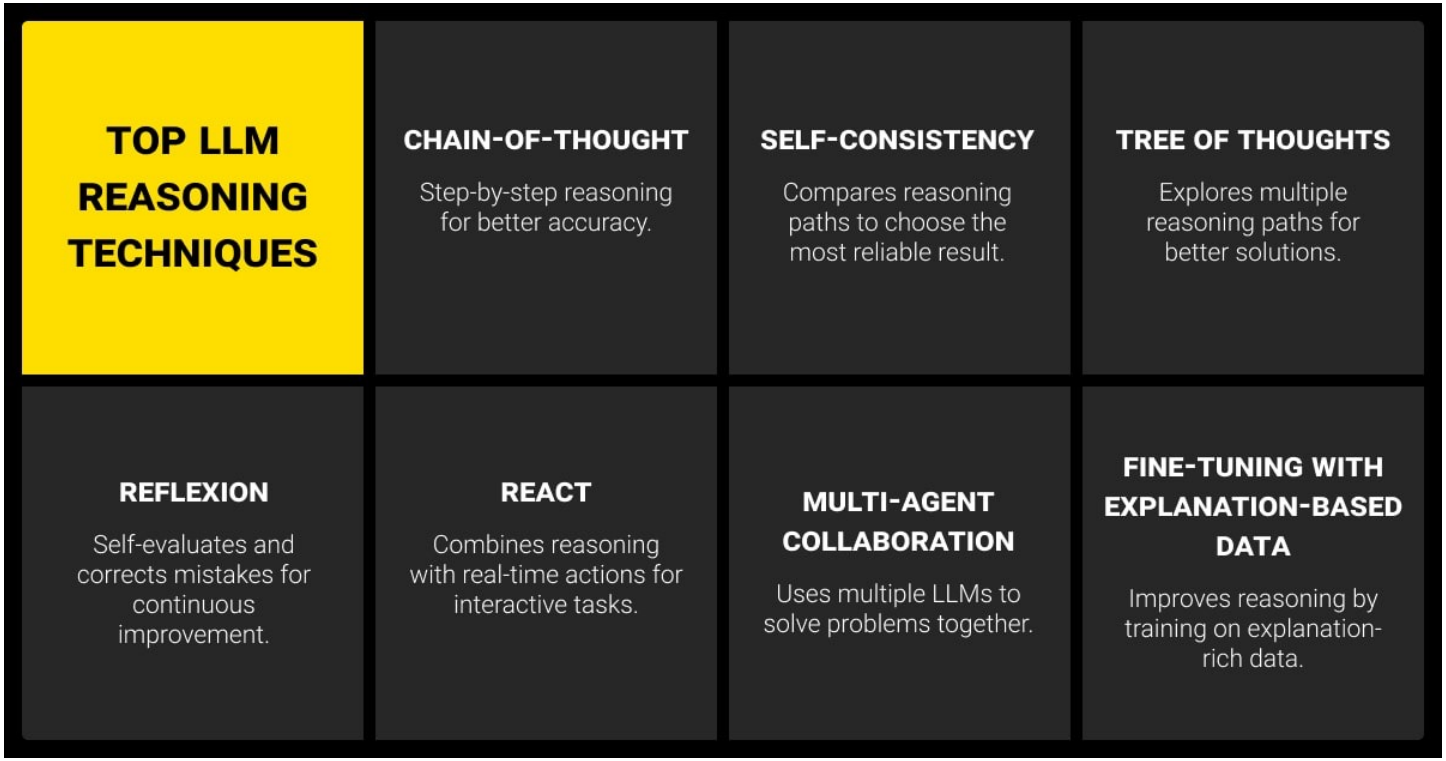
- Prompt B results are significantly higher (0.87 vs. 0.54), meaning it generates responses that better match the ideal answers.
- Refining the prompt to encourage conciseness and relevant details improves model accuracy in responding to customer queries.
- The team decides to adopt Prompt B and further fine-tune it for different query categories.

Logical Consistency

Logical consistency in LLM responses verifies that the output follows a coherent reasoning process, avoids contradictions, and maintains internal validity throughout the response.

To **measure logical consistency**, we assess:

1. **Internal Coherence** – Does the response stay logically sound within itself?
2. **Consistency with Context** – Does it align with the provided facts or input?
3. **Step-by-Step Reasoning Validity** – Does it follow a valid inference process?
4. **Absence of Contradictions** – Does the response contradict itself?



Source: https://labelyourdata.com/articles/llm-reasoning

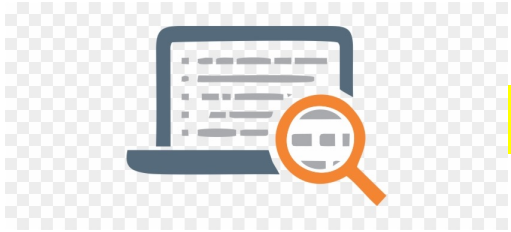
Logical Consistency: Human Evaluation

- **Expert Review (Domain-Specific Consistency Checking)**
  - Experts (e.g., legal professionals, medical doctors or AI researchers) review responses for **correct logical reasoning** and **factual correctness**.
  - **Prompt:** *"Explain how quantum entanglement enables faster-than-light communication."*
  - **Response:** *"Quantum entanglement allows information to be transmitted instantaneously across vast distances."*
  - **Human Reviewer Feedback:** ❌ Incorrect. While entangled particles correlate instantly, **no information is actually transmitted faster than light**. **Comparative Ranking (A/B Testing)**
- **Comparative Ranking (A/B Testing)**
  - Evaluators compare multiple outputs for the same prompt and rank them based on **logical structure** and **reasoning soundness**.
- Example:
  - **Task:** Rank the two responses based on logical consistency.
  - **Response A:** Well-structured, follows cause-effect reasoning.
  - **Response B:** Contains logical jumps, lacks explanation.
  - ✅ **Human raters score Response A higher for consistency.**

Logical Consistency: Human Evaluation

- **Rubric-Based Scoring**
  - Responses are scored on a 1-5 scale for logical consistency based on predefined criteria.
    - Score 5: Fully logical, structured, step-by-step explanation.
    - Score 3: Some logical jumps but mostly coherent.
    - Score 1: Contradictory or illogical reasoning

Human evaluation is **more accurate** for complex logic but **less scalable**.



Automated heuristics are **faster and scalable** but may **miss nuanced errors**.

The best approach might be to **combine both**: **Human evaluators fine-tune heuristics-based models** for long-term improvement.

Logical Consistency: Automated Heuristics

- **Contradiction Detection via NLI (Natural Language Inference)**
  - Use **pre-trained NLI models** (e.g., RoBERTa, BERT) to check if a response contradicts itself or external knowledge.
    - **Statement 1:** *"Photosynthesis requires sunlight."*
    - **Statement 2:** *"Plants can perform photosynthesis in complete darkness."*
    - ✅ **NLI model detects contradiction.**
- **Coherence Scoring with Language Models**
  - Compute **coherence scores** by measuring how logically connected sentences are.
  - Use GPT or BERT embeddings to **compute similarity** between consecutive sentences.
  - If adjacent sentences have **low semantic similarity**, the response may have logical gaps.

Logical Consistency: Automated Heuristics

- **Chain-of-Thought (CoT) Validity Checking**
  - Run the model's **reasoning process** through another LLM or rule-based system to verify **step-by-step correctness**.
    - **Step 1:** Model explains a math problem.
    - **Step 2:** A second LLM re-verifies each reasoning step.
    - ✅ **If all steps are valid, the response is logically consistent.**

- **Knowledge Graph Consistency Checking**
  - Map responses onto a **knowledge graph** to check for logical relationships.
    - A response says *"Elon Musk founded OpenAI."*
    - The knowledge graph verifies that **Musk was a co-founder, not the sole founder**, flagging a **partial inconsistency**.

Diversity and Coverage

- Evaluating the diversity of LLM responses, especially in open-ended generation tasks, is crucial for improving model performance and ensuring useful, creative, and relevant outputs.
- **Quantitative Metrics for Diversity**
  - Entropy-based metrics measures unpredictability. High entropy suggests greater diversity.
    - It can be obtained by examining the distribution of words, sentences, or tokens in the model's responses. A response with varied vocabulary and unpredictable phrasing would have a higher entropy value.
    - Tools like the *nltk* or *scipy* libraries can help calculate entropy.
  - Distinct n-grams: Track how many unique n-grams (combinations of words) appear across multiple responses. The more distinct n-grams, the more varied the model's output.
    - Count distinct 1-grams, 2-grams, or 3-grams from multiple generated outputs.
  - Diversity Score (Distinct-1, Distinct-2): Calculate the ratio of unique 1-grams and 2-grams to the total number of 1-grams and 2-grams generated.
    - For each generated text, calculate the number of unique 1-grams and 2-grams and divide by the total number of 1-grams and 2-grams.

Diversity and Coverage

- **Quantitative Metrics for Coverage**

  - **Perplexity**: Lower perplexity generally indicates less diversity, as the model sticks to more predictable patterns. A higher perplexity might suggest the model is taking risks, creating more diverse output, but possibly at the cost of fluency or coherence.
    - Perplexity can be computed on the generated text to quantify how "risky" or diverse the model's output is compared to standard models.

  - **Semantic Similarity (Cosine Similarity or Embeddings Comparison)**: After generating multiple responses, you can measure the semantic distance between them to understand how different they are in terms of meaning. Lower similarity scores indicate more diverse responses.
    - We can use sentence embeddings (like BERT or GPT embeddings) and calculate cosine similarity between different generated texts.

Hallucinations

- Detecting hallucinations refers to the generation of information that is factually incorrect, misleading, or fabricated
- It requires a combination of strategies that assess factual consistency, cross-reference external structured knowledge, and compare outputs to verified or augmented sources.

Factual Consistency Evaluation

- **Manual Fact-Checking**: One of the simplest ways to detect hallucinations is through human evaluation, where generated text is manually compared to verified, authoritative sources.
  - Ask human evaluators to verify if the LLM output aligns with reliable sources such as academic papers, news articles, or government websites. You can then measure the accuracy rate based on the evaluator's judgment.
- **Automated Fact-Checking Tools**: Use specialized tools to cross-check the model's output with external knowledge bases. This can involve matching generated facts against structured datasets or established fact-checking platforms.
  - Use APIs or frameworks like **ClaimBuster**, **Factmata**, or **Google's Fact-Check Tools** to automate the process of verifying the factual consistency of LLM outputs.
- **Consistency with Known Facts**: For tasks requiring factual accuracy, compare the responses to known facts in a static dataset or curated knowledge repository (e.g., Wikidata, Freebase).
  - Create a test suite of factual questions and check if the model provides consistent and accurate information when asked the same question multiple times.

Hallucinations

Cross-Referencing Structured Knowledge Sources

- **Knowledge Graphs**: Leverage structured knowledge sources like knowledge graphs (e.g., **Wikidata**, **DBpedia**, **YAGO**) that organize facts into interconnected nodes. When an LLM generates a response, you can cross-check the output against the nodes or entities in these graphs to see if the information is accurate and properly linked.
    - Use automated systems to perform entity linking (connecting model output to knowledge graph nodes) and compare the factual accuracy of the entities mentioned.
- **Structured Databases and APIs**: Integrate external databases such as **OpenCage Geocoder** (for geographical data), **IMDb** (for movie information), or **CrossRef** (for academic papers). These databases offer precise, structured knowledge and can be used to validate the accuracy of the model's claims.
    - After generating text, extract key factual assertions (e.g., dates, events, names) and validate these against the structured database using API calls or query mechanisms.
- **Cross-Referencing with News Databases**: If the LLM generates current-event-related information, cross-reference the results with live news databases like **NewsAPI** or archives from credible sources (e.g., Reuters, BBC).
    - Use a set of current events as test queries and compare the LLM’s answers to real-time news data to identify discrepancies.

Hallucinations

Comparing Against Refined-Augmented Outputs

- **Refinement via Expert-Generated or Augmented Data**: Use a human-in-the-loop or automated process to refine the LLM output. This may involve prompting the model to generate a response multiple times and then comparing the results with human-generated content or data from augmented sources (e.g., a combination of pre-processed Wikipedia, research papers, or trusted sources).
    - Compare the LLM's output to human-curated or more robust data sets to identify any factual errors or inconsistencies. Track the number of factual errors across multiple outputs to assess performance.
- **Augmented Response Generation**: Use fine-tuned models or retrieval-augmented generation (RAG) frameworks to improve the factual accuracy of LLM responses by allowing it to pull in real-time data from knowledge sources before generating an answer.
    - Evaluate the model's factual consistency against a manually verified benchmark dataset or factual ground-truth data.
- **External Consistency Checks**: Run the generated text through multiple LLMs with different architectures or fine-tuning data sets to see if the responses align with each other. This can help identify divergent or inaccurate claims.
    - Compare outputs from different models and observe whether they provide consistent or conflicting information on the same topic.

Hallucinations

Using Verification Models

- **Dedicated Fact-Checking Models**: Use specialized verification models like **Factual** (a model designed to determine whether a piece of text is factual) or **ClaimBuster** (which evaluates the veracity of claims). These models can be used in tandem with an LLM to verify whether the generated text is likely to be hallucinated.
    - Feed the LLM outputs into a fact-checking model and measure the accuracy of its classification (e.g., factual, partially factual, or false).
- **Truth-Detection Models**: Leverage truth-detection models or those trained on fact-checking datasets (e.g., **FEVER**, **HoVer**, or **LIAR**). These models specifically detect whether a statement made in a given text is truthful.
    - Use these models to validate the truthfulness of generated responses and measure how many responses are classified as incorrect or misleading.
- **Cross-Verification with External LLMs**: In some cases, running the same prompt through multiple independent models (e.g., GPT-4 vs. T5, or OpenAI vs. another LLM service) can help identify factual discrepancies and highlight hallucinated content.
    - Compare the outputs across different LLMs to identify inconsistencies or hallucinated details.

Hallucinations

Evaluation with Fact-Checking Datasets

- **Public Datasets for Fact-Checking**: Use datasets designed to evaluate fact-checking, such as **FEVER** (Fact Extraction and Verification), **LIAR**, or **SciFact**, which contain claims and their verified true/false statuses.
    - Feed model outputs into these fact-checking datasets to evaluate their accuracy based on pre-labelled data. Metrics like precision, recall, and F1-score can be used to assess how often the model produces correct facts.
- **Comparing Claims Against Source Articles**: For LLMs generating fact-specific content (e.g., answering trivia questions or generating historical information), you can cross-reference the generated claims with the source articles from which those claims originated (e.g., a passage from a scientific paper, news article, or Encyclopedia).
    - Use text comparison methods (e.g., cosine similarity) to measure how closely the generated output matches the facts presented in the original sources.

Adversarial Testing

- **Creating Adversarial Inputs**: Test the LLM by intentionally crafting prompts that are likely to lead to hallucinated responses. This could involve ambiguous or contradictory statements or prompts that are designed to confuse the model.
    - Track how often the model generates hallucinated or incorrect information under these adversarial conditions. This can reveal weaknesses in the model's factual consistency and help improve its robustness.

Prompt Refinement

- **Define the Objective**
  - It's crucial to define the goal or desired outcome of the generated content. This involves specifying what we want the model to achieve (e.g., answering a question, generating creative content, providing recommendations).
    - As example, consider that we want to generate a concise and engaging summary of a news article.
    - **Objective**: "Create a short summary of a news article that highlights the main points and is easy to read."

- **Generate Initial Prompt**
  - The next step is to generate an initial prompt that reflects the defined objective. This prompt will be the starting point for model interactions.
    - **Example**: The initial prompt might be: "Summarize this news article."
  - This is a broad and simple prompt, but it might not provide enough direction for high-quality output.

Prompt Refinement

- **Evaluate Response Quality**
  - After generating the response, evaluate it based on factors like accuracy, clarity, relevance, and adherence to the objective. If the response doesn't meet expectations, identify what needs improvement.
    - **Example of Evaluation**: The model generates a summary, but it's either too long or doesn't capture the key details effectively.
      - **Response**: "*The article discusses various topics including politics, technology, and economics.*"
      - **Evaluation**: The summary is vague and lacks specifics about the news event or main story.
- **Modify Structure, Wording, Constraints**
  - Based on the evaluation, modify the prompt to improve the response. This can include adjusting the wording, adding constraints, or specifying the format more clearly. The goal is to guide the model toward generating a more focused or accurate response.
    - **Example**: To improve the response, you might modify the prompt:
      - Original prompt: "Summarize this news article."
      - Modified prompt: "Summarize the main points of the article in 3-4 sentences, focusing on the key event and its implications."
  - We've added constraints on length and focus, which helps narrow the model's response.

Prompt Refinement

- **Re-test and Optimize**
  - Test the refined prompt and evaluate the new output. If the response improves, continue optimizing by adjusting small details (e.g., refining constraints, adjusting tone, or providing more specific instructions). Repeat this process until the output aligns closely with your objectives.
  - **Example of Re-test**: After refining, you now test the new prompt:
    - **Prompt**: "Summarize the main points of the article in 3-4 sentences, focusing on the key event and its implications."
    - **Response**: "*A recent economic policy shift in the US will impact trade relations. The article highlights concerns from various sectors about the long-term effects. Economists are divided on its potential success, and international trade agreements may also be affected."*
  - ***Evaluation**: This is a more focused, concise summary, and it adheres to the* objective.
- **Optimization**: If further improvements are needed (e.g., more engagement or clarity), refine the prompt again by adjusting the wording or asking for additional details, like "mention the main figures involved" or "include a tone suitable for a general audience."