Uncertainty Quantification for Language Models: A Suite of Scorers and Ensemble Framework

Dylan Bouchard, Mohit Singh Chauhan

Reading Group Presentation

February 6, 2026

Introduction: Purpose & Intuition

Purpose

- **Limitation Addressed:** Current hallucination detection often relies on ”ground truth” (offline evaluation) or external search, which prevents real-time monitoring in production. Existing UQ scores are often unbounded (e.g., perplexity) and hard to interpret.
- **New Settings:** Explores a ”closed-book” setting using a standardized suite of scorers (Black-box, White-box, Judge) and a novel *tunable ensemble* framework.
- **Why it Matters:** Enables safe deployment in high-stakes domains (healthcare, finance) by providing standardized confidence scores to flag unsafe outputs.

Intuition

- **Core Idea/Assumption:** ”Consistency implies Factuality.”
- If an LLM is hallucinating, its internal token probabilities will likely drop (White-box), and if asked the same question repeatedly, its answers will vary semantically (Black-box).
- Combining these signals creates a stronger truthfulness detector than any single method alone.

Potential Applications

Applications Explored by Authors

- Tested on diverse QA Benchmarks: Math (GSM8K), Multiple Choice (CSQA), and Short Answer (PopQA).

Application to Clinical Data (Example)

- *Scenario:* Automated generation of patient discharge summaries or analyzing medical records.
- *Application:* Using UQ scores to flag "low confidence" sections where the model might invent patient history or drug dosages, triggering human review before finalization.

Problem Statement: Hallucination as Binary Classification

The Objective

- Model hallucination detection as a binary classification problem.
- **Definition:** A hallucination is defined as any content that is nonfactual.

Formal Definition

Given a prompt x_i and an LLM response $y_i \in \mathcal{Y}$:

- We define a **Confidence Scorer** $\hat{s} : \mathcal{Y} \to [0, 1]$.
- We predict a hallucination ($\hat{h} = 1$) if the confidence is below a threshold τ:

$$\hat{h}(y_i; \cdot, \tau) = \mathbb{I}(\hat{s}(y_i; \cdot) < \tau)$$

- **Note:** $\hat{h} = 1$ implies a hallucination; $\hat{h} = 0$ implies factual.

The Key Challenge (Closed-Book Setting)

- **Ideal Ground Truth (h):** Requires comparing y_i against a correct reference answer y_i^*.
- **Real-World Constraint:** In production generation-time, y_i^* is **not available**.

Black-Box UQ: Overview

Core Intuition:

- Exploit the stochastic nature of LLMs.
- Generate m candidate responses $\tilde{y}_i = \{\tilde{y}_{i1}, ..., \tilde{y}_{im}\}$ from the same prompt x_i.
- Measure **Semantic Consistency** between the original response y_i and candidates $\tilde{y}_i$.
- Demo

We will introduce 5 specific scorers:

1. Exact Match Rate (EMR)
2. Non-Contradiction Probability (NCP)
3. BERTScore Confidence (BSC)
4. Normalized Cosine Similarity (NCS)
5. Normalized Semantic Negentropy (NSN)

1. Exact Match Rate (EMR)

Definition

- Measures the proportion of candidate responses that are identical to the original response.
- Useful for tasks with unique, closed-form answers (e.g., math problems).

Formula Given original response y_i and candidates $\tilde{y}_{ij}$:

$$EMR(y_i; \tilde{y}_i, x_i) = \frac{1}{m} \sum_{j=1}^{m} \mathbb{I}(y_i = \tilde{y}_{ij})$$

where $\mathbb{I}$ is the indicator function.

2. Non-Contradiction Probability (NCP)

Definition

- Uses a Natural Language Inference (NLI) model to check logical consistency.
- Measures how often the original response does **not** contradict the candidates.

Formula Let $\eta(A, B)$ be the probability that A contradicts B:

$$NCP(y_i) = 1 - \frac{1}{m} \sum_{j=1}^{m} \frac{\eta(y_i, \tilde{y}_{ij}) + \eta(\tilde{y}_{ij}, y_i)}{2}$$

Higher score → Fewer contradictions → Higher confidence.

Detailed breakdown

1. y_i is the original response; $\tilde{y}_{ij}$ is the j-th sampled candidate response (out of m samples).
2. $\eta(A, B)$ is the NLI model's probability that text A *contradicts* text B.
3. **Bidirectional averaging:** we use $\frac{\eta(y_i, \tilde{y}_{ij}) + \eta(\tilde{y}_{ij}, y_i)}{2}$ because NLI is not symmetric (premise → hypothesis matters).
4. $1 - (\cdot)$ converts average contradiction probability into a *non-contradiction* confidence score: closer to 1 means higher consistency.

Why NLI is Asymmetric?

The Concept: Directionality of Entailment

- Natural Language Inference (NLI) determines if a *Premise* implies a *Hypothesis*.
- **Asymmetry:** The relationship is often one-way. $A \implies B$ does not guarantee $B \implies A$.
- *Key Insight:* Information flow usually goes from **Specific** to **General**.

Direction 1: Specific → General

Premise (y_i): ”A black cat is sleeping.”
Hypothesis ($\tilde{y}_{ij}$): ”An animal is sleeping.”
Result: Entailment
(A cat is necessarily an animal.)

Direction 2: General → Specific

Premise ($\tilde{y}_{ij}$): ”An animal is sleeping.”
Hypothesis (y_i): ”A black cat is sleeping.”
Result: Neutral
(The animal could be a dog!)

Implication for the Paper (NCP Metric)

- Relying on a single direction (e.g., only $y_i \rightarrow \tilde{y}_{ij}$) may falsely classify vague responses as ”consistent” with specific ones.
- **Solution:** The paper uses **Bidirectional Averaging** to ensure robust semantic consistency checks :

3. BERTScore Confidence (BSC)

Definition

- Uses contextualized word embeddings (BERT) to measure soft similarity.
- Captures semantic overlap even if phrasing differs.

Formula Computes the average BERTScore F1 between the original response and all candidates:

$$BSC(y_i) = \frac{1}{m} \sum_{j=1}^{m} \text{BERTScoreF1}(y_i, \tilde{y}_{ij})$$

- Calculates similarity via token-level greedy matching in embedding space.

4. Normalized Cosine Similarity (NCS)

Definition

- Uses a Sentence Transformer to map entire responses to vector embeddings $V(\cdot)$.
- Measures global semantic similarity in the embedding space.

Formula

$$NCS(y_i) = \frac{1}{2m}\sum_{j=1}^{m}\frac{V(y_i)\cdot V(\tilde{y}_{ij})}{\|V(y_i)\|\|V(\tilde{y}_{ij})\|} + \frac{1}{2}$$

- **Normalization:** Cosine similarity ranges $[-1, 1]$. The term $\frac{1}{2}(\cdot) + \frac{1}{2}$ maps it to $[0, 1]$.

5. Normalized Semantic Negentropy (NSN)

1. Foundation: Semantic Entropy (SE)

Concept: Meaning over Wording

- Traditional entropy on raw strings overestimates uncertainty when the model generates diverse phrasings for the *same answer* (e.g., "Paris" vs. "It is Paris").
- **Mechanism:**
  1. Cluster all responses $\{y_i, \tilde{y}_{i1}, ..., \tilde{y}_{im}\}$ based on **bi-directional entailment** (NLI).
  2. Responses in the same cluster share the same semantic meaning.
  3. Calculate entropy over the distribution of these semantic clusters $\mathcal{C}$:

$$SE(y) = -\sum_{C \in \mathcal{C}} P(C) \log P(C)$$

5. Normalized Semantic Negentropy (NSN)

2. The Proposed Metric: NSN (Confidence Score)

Normalization & Inversion

- **Problem:** $SE \in [0, \infty)$, making it hard to use in an ensemble.
- **Solution:** Normalize by max entropy $\log(m+1)$ and invert to represent *confidence*:

$$NSN(y_i) = 1 - \frac{SE(y_i)}{\log(m+1)}$$

- *Interpretation:* $1 \implies$ All responses mean the same thing (High Confidence).

Evaluation of Black-Box Methods

Comparison & Critique (Related Work)

- **Exact Match / Repetition:**
  - *Pros:* Simple to compute.
  - *Cons:* Penalizes minor phrasing differences (e.g., "The cat" vs "A cat"). Too stringent for open-ended generation.
- **N-gram Metrics (ROUGE/BLEU):**
  - *Cons:* Highly sensitive to word order; fail to detect semantic equivalence when phrasing varies significantly.
- **Embedding & NLI Methods (NCS, NSN, NCP):**
  - *Pros:* Can detect semantic similarity across different phrasings.
  - *Performance:* NLI-based methods (NSN, NCP) generally outperform others, especially in capturing logical consistency.
  - *Cons:* Higher computational cost (requires running NLI models).

White-Box UQ: Overview

Core Intuition

- Leverage the internal token probabilities (logits) of the LLM.
- Does not require sampling multiple responses (faster than Black-box).
- Requires access to model internals (not always available for API models).
- Demo

1. Length-Normalized Token Probability (LNTP)

Definition

- Computes the geometric mean of the probabilities of all tokens in the response.
- Equivalent to the exponential of the average log-probability.

Formula Given response y_i with tokens $\{t_1, ..., t_L\}$:

$$LNTP(y_i) = \left(\prod_{k=1}^{L} P(t_k | t_{<k}) \right)^{\frac{1}{L}}$$

- Values are naturally in $[0, 1]$.

2. Minimum Token Probability (MTP)

Definition

- Uses the probability of the *least likely* token in the sequence as the confidence score.
- Based on the intuition that a single highly uncertain token can invalidate the entire response (weakest link).

Formula

$$MTP(y_i) = \min_{t \in y_i} P(t)$$

Evaluation of White-Box Methods

Comparison & Critique (Related Work)

- **Raw Neg-Log Probability / Perplexity:**
  - *Cons:* Unbounded ($[0, \infty)$), hard to interpret as a standalone confidence score.
- **Joint Probability (Response Improbability):**
  - *Cons:* Penalizes longer responses. A long correct answer will have lower probability than a short incorrect one.
- **Length-Normalized (LNTP):**
  - *Pros:* Bounded in $[0, 1]$, easy to interpret, robust to length variations.
- **General Limitation:**
  - Requires white-box access, which is impossible for many closed APIs (e.g., standard ChatGPT web interface).

LLM-as-a-Judge Scorer

Methodology

- Concatenate the Question + Generated Answer.
- Ask an LLM (can be the same or a different model) to evaluate correctness.
- **Prompt Strategy:** Explicitly instruct the model to output a confidence score between 0 and 100.

Standardization

- Normalize the output (0-100) to $[0, 1]$ to be consistent with other scorers.
- Appendix Eg

Evaluation of LLM-as-a-Judge

Critique (Related Work)

- **Self-Reflection:**
  - *Concept:* Asking the model "Are you sure?" (P(Correct)).
  - *Pros:* Simple, requires no ground truth.
- **Model Bias:**
  - Large models generally make better judges.
  - A model's accuracy on a task correlates with its ability to judge that task.
- **Panel of LLMs (PoLL):**
  - *Insight:* Using a panel of smaller LLMs can sometimes outperform a single large judge and reduce intra-model bias.

Tunable Ensemble Scorer

Goal:

- Diverse strengths of various uncertainty signals
- Provides a flexible and tunable mechanism that allows practitioners to optimize the importance of specific components tailored to their unique use cases and datasets

Setup

- Prompt x_i, response y_i, and candidates $\tilde{y}_i$.
- K individual scorers $\hat{s}_k(y_i; \tilde{y}_i, x_i)$.

Ensemble confidence For an original response y_i, the ensemble confidence score is defined as:

$$\hat{s}(y_i; \tilde{y}_i, x_i, w) = \sum_{k=1}^{K} w_k \, \hat{s}_k(y_i; \tilde{y}_i, x_i)$$

Constraints on weights

$$\sum_{k=1}^{K} w_k = 1, \quad w_k \in [0, 1]$$

Ensemble Tuning: Optimization Strategies

Prerequisite: Graded Dataset

- Tuning requires a sample of n prompts with responses y and ground-truth hallucination labels $h(y; y^*)$ (derived from human grading or automatic rules).

Strategy A: Threshold-Agnostic Optimization (AUROC)

- **Step 1:** Optimize weights w^* to maximize a separation metric $\mathcal{S}$ (e.g., AUROC).

$$w^* = \arg\max_{w \in \mathcal{W}} \mathcal{S}(\hat{s}(y; \cdot, w), h(y; \cdot))$$

- **Step 2:** Tune threshold τ^* separately after fixing weights.

Strategy B: Threshold-Aware Optimization (F1-Score)

- Jointly optimize weights w and threshold τ to maximize a specific decision metric $\mathcal{B}$ (e.g., F1-score).

$$w^*, \tau^* = \arg\max_{w, \tau} \mathcal{B}(\hat{h}(y; \cdot, w, \tau), h(y; \cdot))$$

- *Implementation:* Authors use Optuna for this hyperparameter search.

Experimental Setup: Tasks, Models, and Sampling

- **Goal:** Evaluate uncertainty (UQ) scorers for detecting whether an LLM answer is *correct* vs *incorrect* (binary label).
- **Benchmarks (6 datasets)** grouped by answer format:
  - **Math (numeric):** GSM8K, SVAMP
  - **Multiple-choice:** CSQA, AI2-ARC
  - **Short-answer:** PopQA, NQ-Open
- **Evaluation scale:** 1000 prompts per dataset; 4 LLMs × 6 datasets = 24 LLM–dataset scenarios.
- **Sampling protocol (per prompt):**
  - Generate 1 *original response* y_i
  - Sample $m = 15$ *candidate responses* $\tilde{y}_{i1}, \ldots, \tilde{y}_{im}$ at temperature 1.0

Evaluation Metrics and Cross-Validation Protocol

- Binary ground truth: $h_i \in \{0, 1\}$ (correct vs incorrect), produced via task-specific graders.
- **Threshold-agnostic metric: AUROC**
  - Measures ranking quality of scores $\hat{s}_i$ vs labels h_i.
  - **5-fold CV:** train/tune on 4 folds, evaluate AUROC on the held-out fold; report mean across folds.
- **Threshold-optimized metric: F1-score**
  - Choose threshold τ to convert scores to predictions:

$$\hat{h}_i(\tau) = \mathbb{I}\{\hat{s}_i \geq \tau\}.$$

  - **5-fold CV:** pick τ on tuning folds (grid search), then compute F1 on the held-out fold.
- **Selective prediction: Filtered Accuracy@τ**
  - Compute accuracy only on examples with $\hat{s}_i \geq \tau$; sweep $\tau \in \{0, 0.1, \ldots, 0.9\}$.

Key Results: AUROC and F1 Across 24 Scenarios

- **No universal best single scorer:** best-performing family depends on the LLM–dataset scenario.
- **Ensembling helps in most settings:**
  - Ensemble often outperforms individual components in AUROC and in F1 after threshold tuning.
  - Interpretation: different scorers capture complementary uncertainty signals (agreement vs likelihood vs judge feedback).
- **Black-box vs white-box vs judge:** performance varies by task type
  - NLI-based agreement tends to be strong among black-box methods.
  - White-box log-prob based scores are competitive but not always dominant.
  - Judge performance can be task-dependent (e.g., math vs short-answer).
- **In-domain tuning:** ensemble weights are tuned per LLM–dataset pair (use-case specific deployment).

Practical Takeaway: Using Scores for Safer Deployment

- **Filtered Accuracy@τ rises with τ:**
  - As we keep only high-confidence answers, empirical accuracy increases (often near-monotonic).
  - Enables **selective generation**: answer automatically when confident; otherwise defer to retrieval, tools, or humans.
- **Operational workflow (recommended):**
  1. Choose a scorer family (or ensemble) appropriate for your setting (black-box / white-box / judge).
  2. Tune ensemble weights on a labeled validation set (AUROC for ranking; or F1 if a fixed decision is required).
  3. Choose threshold τ based on desired precision–recall trade-off and cost of errors.
  4. Deploy: accept if $\hat{s} \geq \tau$; else abstain / request more evidence / re-query.
- **Message:** uncertainty scoring is not only an offline metric—it directly improves real-world reliability via abstention.