From Input Perception to Predictive Insight: Modeling Model Blind Spots Before They Become Errors

Maggie Mi[1] **Aline Villavicencio**[2,3,4] **Nafise Sadat Moosavi**[1]
[1]University of Sheffield [2]University of Exeter [3]The Alan Turing Institute [4]UFRN, Brazil
{zmi1, n.s.moosavi}@sheffield.ac.uk
a.villavicencio@exeter.ac.uk

Abstract

Language models often struggle with idiomatic, figurative, or context-sensitive inputs, not because they produce flawed outputs, but because they misinterpret the input from the outset. We propose an input-only method for anticipating such failures using token-level likelihood features inspired by surprisal and the Uniform Information Density hypothesis. These features capture localized uncertainty in input comprehension and outperform standard baselines across five linguistically challenging datasets. We show that span-localized features improve error detection for larger models, while smaller models benefit from global patterns. Our method requires no access to outputs or hidden activations, offering a lightweight and generalizable approach to pre-generation error prediction.

https://github.com/mi-m1/input_perception

1 Introduction

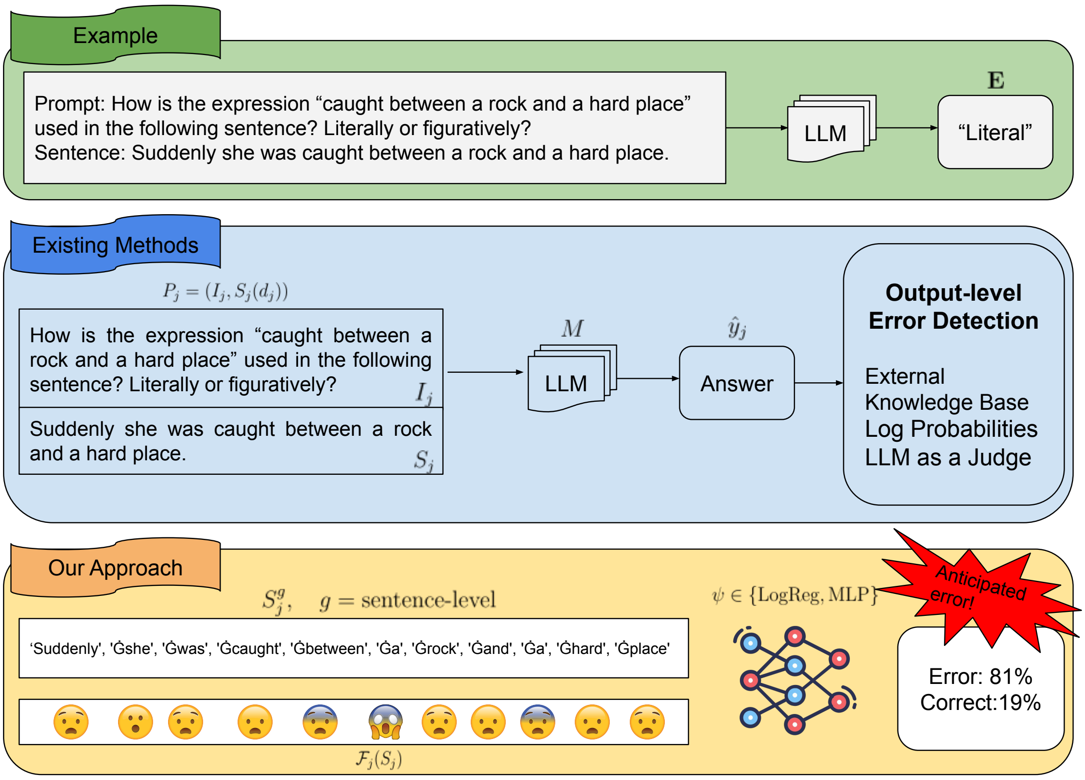
Model failures in language understanding do not always stem from incoherent outputs. Instead, they may originate earlier in the processing pipeline, from how the model internally interprets the input. When meaning depends on context-sensitive constructions, such as idiomatic or metaphorical expressions, the model may produce a plausible response that is nevertheless grounded in a misreading of the input. These cases reveal a blind spot in the model's comprehension, where high-confidence generation masks a fundamental interpretive error. This raises a critical question: *Can we anticipate such errors before the model generates a response, purely by examining how it internally processes the input?* Prior work has shown that models tend to perform better on inputs they assign higher overall likelihoods (Ohi et al., 2024; McCoy et al., 2024), suggesting that token-level probabilities may encode latent signals of confidence or uncertainty. However, most existing approaches reduce this likelihood information to a global scalar, such as perplexity, and do not examine how fine-grained variations across the input might reflect deeper patterns of misalignment. Moreover, nearly all existing techniques for error or uncertainty estimation rely on decoding-time cues, such as logits (Belrose et al., 2023), entropy (Pereyra et al., 2017), or sampling variance. Our method, however, is complementary: it anticipates failure at the input stage without consulting the output.

We propose an input-driven framework for anticipating language model (LM) errors by analyzing the structure of the likelihood surface over input sequences. Our approach is motivated by the Uniform Information Density (UID) hypothesis (Jaeger and Levy, 2006; Jaeger, 2010), which views language as a signal optimized to distribute information evenly across an utterance.

In practice, information density fluctuates due to grammatical, discourse, and pragmatic factors (Levy, 2013; Genzel and Charniak, 2002; Xu and Reitter, 2016), and forms information contours that reflects meaningful variations of contextual information (Tsipidi et al., 2024). Similarly, non-literal language, e.g., idioms, metaphors, metonymy, violates compositional expectations and disrupts local information uniformity. For LMs, such irregularities appear as perturbations in the likelihood landscape. We hypothesize that these points of instability, where predictive expectations diverge from natural variability, offer useful signals for anticipating model errors.

Although our framework applies to many kinds of context-sensitive meaning, we target idioms, metaphors, and metonymy because they stress contextual interpretation and have been repeatedly shown be be challenging for LLMs: misunderstandings of context can flip the label from literal to figurative (or vice versa) and yield large errors. Recent studies show that even strong models of-

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 34317–34330
November 4-9, 2025 ©2025 Association for Computational Linguistics


Prompt: How is the expression “caught between a rock and a hard place” used in the following sentence? Literally or figuratively?
Sentence: Suddenly she was caught between a rock and a hard place.

How is the expression “caught between a rock and a hard place” used in the following sentence? Literally or figuratively?

Suddenly she was caught between a rock and a hard place.

'Suddenly', 'Ġshe', 'Ġwas', 'Ġcaught', 'Ġbetween', 'Ġa', 'Ġrock', 'Ġand', 'Ġa', 'Ġhard', 'Ġplace'

Figure 1: An illustration of an LLM failure on a non-compositional semantics task (top panel). Standard error detection methods typically work by checking the generated output afterward, often requiring extra resources such as a Judge LLM (middle panel). In contrast, our method applies information-theoretic measures directly to the input sentence (bottom panel). In this case, our model estimated an 81% likelihood that the LLM would fail on this example, and indeed, it did.

ten fail to leverage context for these distinctions (Mi et al., 2025; Phelps et al., 2024), making these tasks an incisive testbed for input-side error prediction. A further advantage is observability: these datasets come with explicit span annotations for the potentially problematic phrases (e.g., the idiom or the metaphoric span). This lets us localize input-likelihood features to known points of interpretive risk and investigate whether the model's own input-side signals anticipate mistakes more accurately than coarse global heuristics.

Our results show that token-level likelihood features, without access to output logits or hidden states, significantly improve error detection, particularly for smaller models, and outperform established baselines such as log probability, max token confidence, or Oddballness.

While our span-localized features leverage task-informed linguistic structure, our sentence-level features generalize across settings and suggest a broader insight: that the internal likelihood surface over the input encodes rich, interpretable signals of model comprehension. This opens the door to future work on black-box risk estimation that operates before generation, using only how the model “reads” the input as a basis for identifying when it is likely to fail.

Contributions (1) We propose an input-driven framework for anticipating language model errors using token-level likelihood features, without relying on outputs or internal activations. (2) We develop both global and span-localized uncertainty features; while the latter are guided by task-specific linguistic structure, the global features are broadly applicable and offer a path toward dynamic localisation. (3) We demonstrate the effectiveness of this approach across five linguistically grounded datasets, showing substantial gains over standard input-likelihood heuristics, particularly for smaller models (1B-3B parameters).

2 Related Work

Literal vs. Figurative Understanding The task of detecting metaphors, idioms, and metonymy involves determining whether a linguistic expression is used figuratively or literally. Despite substantial progress, this remains a challenging problem for language models (Phelps et al., 2024; He et al., 2024; Yang et al., 2024; Tian et al., 2024; Steen et al., 2010). A key difficulty lies in models' inability to effectively leverage context for disambiguation. For example, Mi et al. (2025) show that even under controlled, contrastive evaluation settings, large language models struggle to use contextual cues to distinguish figurative from literal interpretations.

Kabbara and Cheung (2022) demonstrate that Transformer-based models often rely on superficial lexical or structural cues, rather than engaging in deeper pragmatic reasoning. This finding aligns with our hypothesis that many model errors arise not from output generation, but from internal misinterpretation of the input. If models rely on superficial cues rather than deeper semantic reasoning, then fine-grained input likelihood patterns, such as local surprisal spikes, may offer a more faithful signal of confusion or misalignment. By analyzing these token-level signals, our approach seeks to uncover where and how models exhibit shallow comprehension, especially in linguistically complex regions.

Surprisal and Psycholinguistics Signals Surprisal, rooted in Shannon's information theory (Shannon, 1948), measures the unexpectedness of a word in context, with less predictable words imposing greater processing difficulty on humans (Hale, 2001; Levy, 2008). A large of body of literature has shown that surprisal values estimated from neural LMs (e.g., LSTMs and Transformers) predict human reading times (Oh and Schuler, 2023; Wilcox et al., 2023; Pimentel et al., 2023; Rambelli et al., 2023; Goodkind and Bicknell, 2018).

Other contextual predictors have also been explored. Entropy captures uncertainty about upcoming words, reflecting the number of plausible continuations in context (Futrell et al., 2020). Pointwise Mutual Information (PMI) (Fano, 1961; Church and Hanks, 1990) quantifies word associations, and has long been used in modeling lexical co-occurrence and semantic similarity. Together, these measures capture token-level dimensions of processing difficulty: unexpectedness (surprisal), contextual uncertainty (entropy), and pairwise association strength (PMI). While primarily applied to model human comprehension, recent work reframes these signals as useful diagnostics for LMs themselves (Opedal et al., 2024).

Error detection A broad body of work has explored error detection and uncertainty in language models, often focusing on output-driven signals. Selective prediction (Gu and Hopkins, 2023), entropy-based methods (Shorinwa et al., 2024; Wu et al., 2025), and token-level uncertainty estimation (Ma et al., 2025) estimate uncertainty during or after generation, by relying on output probabilities or decoding dynamics (Wu et al., 2025). Other approaches probe model internals, e.g., hidden states (Burns et al., 2024; Azaria and Mitchell, 2023; Li et al., 2023; Zou et al., 2025), attention patterns (Chuang et al., 2024), or gradients, to diagnose understanding (Ashok and May, 2025; Vig and Belinkov, 2019; Chefer et al., 2021). While effective, these methods require access to internal representations or model outputs, limiting their applicability to open-source or non-black-box settings.

In contrast, our approach estimates error risk before any generation occurs, using only token-level input likelihoods. This purely input-driven method allows us to infer model comprehension externally, making it compatible with black-box or API-only models. Our work is also motivated by studies showing that models perform better on inputs with higher overall likelihoods (Ohi et al., 2024; McCoy et al., 2024), and by techniques like Oddballness (Gralinski et al., 2025), which flag anomalous inputs via likelihood deviations. These findings suggest that the input likelihood surface carries important signals of conceptual instability, particularly in linguistically complex regions and thus, offers a lightweight and generalizable path for proactive error detection.

3 Method

3.1 Modeling Comprehension through Input Likelihood Structure

Our approach is based on the hypothesis that the likelihoods a language model assigns to tokens within an input sequence encode latent signals of its internal comprehension. Specifically, we posit that surprisal and related features derived from input token probabilities can reveal regions of uncertainty or misalignment that precede downstream errors. This view aligns with psycholinguistic findings showing that humans experience increased processing difficulty at points of high surprisal (Hale, 2001; Smith and Levy, 2013), and recent evidence suggesting similar interpretive dynamics in LLMs (Ohi et al., 2024).

Rather than analyzing outputs or decoder activations, we focus exclusively on the model's perception of the input. Given a dataset $\mathcal{D}$ comprising instances $d_j \in \mathcal{D}$, and we construct a prompt, P_j, composed of two components:

$$\mathcal{P}_j = (Q_j, S_j(d_j)) \quad (1)$$

where Q_j denotes the task instruction specifying the phenomenon relevant to task j and S_j denotes the contextual sentence for instance d_j.

We extract features from the likelihood distribution over S_j, treating this distribution as a proxy for the model's internal interpretation of the input.[1] These features are computed either over the entire sentence (global) or over spans informed by linguistic structure (localized).

3.2 Measures

We define a set of information-theoretic measures, as:

$$\Phi = \{\text{SPR}, \text{H}, \text{CWS}, \text{CIS}\}$$

Each measure, $k \in \Phi$, captures a distinct facet of model's predicative behaviour over various granularity (e.g., S_j) of the input. These features require no task-specific information and are broadly applicable across inputs.

From these metrics, we derive sentence-level features by aggregating values across the context sentence (S_j) (e.g., mean surprisal, maximum entropy). Together, these features provide complementary perspectives on the model's internal belief state, capturing token-level fit, uncertainty sharpness, confidence calibration, and contextual salience, and serve as interpretable signals for anticipating downstream error.

Surprisal Surprisal measures the unexpectedness of a token t_i, defined as the negative log-probability of the token given its preceding context:

$$\text{SPR}(t_i) = -\log_2 P(t_i \mid t_{<i})$$

Entropy. Shannon entropy reflects the model's uncertainty over the next token distribution at each position. Unlike surprisal, which is token-specific, entropy quantifies the overall spread of the model's prediction:

$$H(t_i) = -\sum_{w \in V} P(w \mid t_{<i}) \log_2 P(w \mid t_{<i})$$

Confidence-Weighted Surprisal (CWS) We propose a variant of surprisal that incorporates a penalty for low-confidence or diffuse next-token distributions. CWS augments surprisal with a KL divergence term measuring deviation from an idealized, peaked distribution:

$$\text{CWS}(t_i) = -\log_2 P(t_i \mid t_{<i}) + \gamma \cdot D_{\text{KL}}(P \parallel Q)$$

Here, P is the model's predicted distribution, and Q is a reference distribution that assigns 0.9 probability to the observed token and distributes the remaining 0.1 uniformly. The penalty weight γ controls sensitivity to this divergence:

$$D_{\text{KL}}(P \parallel Q) = \sum_{t \in V} P(t) \left[\log_2 P(t) - \log_2 Q(t)\right]$$

We introduce CWS to explicitly combine token correctness and sharpness of the belief, since there are settings where a model assigns moderately high probability to the observed input tokens but with a diffuse overall distribution (high entropy), thus, indicating low semantic commitment. This hybrid aims to penalise such diffuse predictions more directly.

Contextual Influence Score (CIS). CIS measures the incremental informatoin that t_i contributes to predicting t_{i+1}:

$$\begin{aligned}\text{CIS}(t_i) = {} & \log_2 P(t_{i+1} \mid t_{\leq i}) \\ & - \log_2 P(t_{i+1} \mid t_{<i})\end{aligned}$$

Equivalently, it is a conditional PMI term. Operationally, the second term is computed by rescoring t_{i+1} under the shortened prefix $t_{<i}$ (i.e.,"without t_i"). Thus, CIS measures the conditional PMI between t_i and t_{i+1} given the prefix, and quantifies the incremental predictive contribution of t_i during autoregressive inference.

3.3 Features Localized to Challenging Input Regions

While sentence-level features summarize the model's input-conditioned belief profile, many reasoning failures stem from localized comprehension difficulties. In tasks with semantically complex constructions, the model must integrate context and resolve ambiguity within a confined span (e.g., the idiomatic phrase in S_j: $S_{j,\text{expr}}$). We hypothesize that token-level likelihood patterns in such

[1] While we do not explicitly design or optimize prompts, we acknowledge that the distribution over S_j is shaped by I_j due to the model's autoregressive architecture.

spans can more precisely indicate potential errors than global averages. When the boundaries of these challenging regions are known or can be inferred (e.g., via dataset annotation), we compute localized features informed by linguistic theory. We operationalize three linguistically motivated hypotheses that link likelihood dynamics to characteristic comprehension challenges.

Fixedness of Idioms (FOI). Idiomatic expressions are often syntactically rigid and lexically constrained (Chafe, 1968; Fraser, 1970). Once the idiom is initiated, its subsequent tokens become increasingly predictable. In surprisal terms, we expect a decreasing pattern. We capture this with two features:

Monotonic Decrease: A binary feature indicating whether information decrease throughout the span:

$$\text{Decreasing}_{\text{expr}} = \mathbb{1}\Big(\forall\, i < j \in S_{j,\text{expr}},\; k(w_i) > k(w_j)\Big)$$

Surprisal Spikes: The number of local surprisal maxima in the span, reflecting unpredictability:

$$N_{\text{spikes}} = \Big|\big\{\, i \in S_{j,\text{expr}} \,\big|\, k(w_i) > k(w_{i-1}) \wedge k(w_i) > k(w_{i+1}) \,\big\}\Big|$$

Selectional Preference Violations (SPV). Metaphorical constructions often involve semantic mismatches between verbs and their arguments (Wilks, 1975, 1978), which may lead to sharp information transitions. We define **Boundary Shift**, which represents the change in information from the final token in the span to the token immediately following it:

$$\Delta_{\text{boundary}} = k(w_{\text{post}}) - k(w_{\text{end}})$$

where w_{end} is the last token of the span, and w_{post} is the following token.

High Context Information (HCS). Comprehension failures often correlate with local information peaks. We test whether the model's highest uncertainty is localized within the span using **Peak-in-Span Indicator:**

$$\delta(x) = \mathbb{1}(i^* \in S_{j,\text{expr}}), \quad i^* = \arg\max_i k(w_i)$$

We also design additional features that are inspired by there linguistic theories (see Appendix A. Collectively, these localized features are designed to complement sentence-level likelihood summaries by providing finer-grained signals of conceptual instability within semantically challenging parts of the input.

Granularities. To capture cues for distinguishing literal from figurative usage, we compute features at four granularities. Sentence-level applies metrics to the full sentence. Expression-level restricts to the idiomatic span. Boundary-level focuses on words immediately flanking the idiom. Context-level uses the surrounding sentence with the idiom removed. These complementary views isolate internal, local, and contextual signals shaping interpretation.

3.4 Feature Set

For each tokenwise metric defined in Section 3.2 (e.g., surprisal, entropy, CIS, CWS), and for each example j with prompt $\mathcal{P}_j$ tokenized as $\mathbf{x}_j = (x_{j,1}, \ldots, x_{j,m_j})$, we derive features at different levels of granularity. With Φ denoting the set of measures and $\mathcal{G} = \{$*sentence-level*, *expression-level*, *boundary-level*, *context-level*$\}$ the granularities. For any region $g \in \mathcal{G}$, let $I_j^g \subseteq \{1, \ldots, m_j\}$ be the corresponding index set of tokens (e.g., all tokens for *sentence-level*, the annotated span for *expression-level*, etc.). We aggregate over a set of operators $\mathcal{A} = \{\text{mean}, \text{maxmin}, \text{std}\}$:

$$f_{j,k,g,a} = \begin{cases} \frac{1}{|I_j^g|} \sum_{t \in I_j^g} k(x_{j,1:t}), a = \text{mean}, \\ a_{t \in I_j^g} k(x_{j,1:t}), a \in \{\text{max}, \text{min}, \text{std}\}. \end{cases}$$

Thus, our feature set for our linguistically targeted setting is:

$$\mathcal{F}_j = \big\{\, f_{j,\,k,\,g,\,a} \,\big|\, k \in \Phi,\; g \in \mathcal{G},\; a \in \mathcal{A} \,\big\}. \quad (2)$$

Sentence-level restriction. When restricting to sentence-level evaluations, we fix the granularity to sentence-level and apply the mean and maximum aggregations across the sentence. Formally,

$$\mathcal{F}_j^{\text{sent}} = \big\{\, f_{j,\,k,\,S_j,\,\text{mean}},\; f_{j,\,k,\,S_j,\,\text{max}} \,\big|\, k \in \Phi \,\big\}. \quad (3)$$

4 Experimental Setup

4.1 Datasets

We utilize five benchmark datasets spanning three distinct types of non-literal language understanding: idiomaticity, metaphor, and metonymy.

DICE (Mi et al., 2025) is an idiomaticity detection dataset that preserves the fixed lexical and syntactic form of idioms across both literal and figurative uses. Unlike prior datasets that alter idiom structure in literal cases, DICE forces models to rely on context rather than form to disambiguate meaning. MOH-X (Mohammad et al., 2016) and TRoFi (Birke and Sarkar, 2006) are datasets for verbal metaphorical/literal identification. We use Task 14 *Reference via Metonymy* of PUB (Sravanthi et al., 2024) (henceforth, PUB 14), which are challenging cases where named entities refer not to themselves but to entities closely associated with them (e.g., "Washington" referring to the U.S. government). Similarly, ConMeC (Ghosh and Jiang, 2025) focuses on metonymy of common, high-frequency nouns (such as ‘glass’ for ‘wine’).

4.2 Evaluation Paradigm

Given a LLM M, we use zero-shot prompts to assess the knowledge that the model has learned during training, rather than its ability to adapt to the task using in-context learning. Since understanding context is an inherent aspect of language ability, and not a downstream task, we aim to evaluate the model in its unaltered state. The only variation in the task instruction is the specification of the particular phenomenon relevant to each task. We provide our prompts for each task in Appendix B.1. Thus, the model prediction is then given by:

$$\hat{y_j} = M(\mathcal{P}_j) \tag{4}$$

The accuracy of the model for each instance is evaluated by comparing $\hat{y_j}$with the gold label y_j. We provide the accuracy of the models evaluated in Appendix C.1.

$$\text{Accuracy} = \frac{1}{|\mathcal{D}|}\sum_{j=1}^{|\mathcal{D}|}\mathbb{I}[\hat{y}_j = y_j] \tag{5}$$

Let $\mathbf{e} = (e_1, \ldots, e_{|\mathcal{D}|})$ denote the vector of error labels, where $e_j = \mathbf{1}[\hat{y}_j \neq y_j]$.

4.3 Classifiers

We fit logistic regression and an MLP to map $\mathcal{F}_j$ to an error probability $\hat{p}_j$. For binary decisions we use a threshold τ (Appendix B.3). Further details are in Appendix B.3.

4.4 Baselines

Typically, methods such as log probability and max token probability have been used as signals for error detection. Thus, we fit our logistic regression model and MLP model on: log probability, max token probability and oddballness as baselines (see Appendix B.5).

5 Results

5.1 Can input likelihood features signal model failures?

Table 1 reports F1 for error detection across models and datasets, comparing simple input-likelihood heuristics (mean log likelihood, mean max token probability, Oddballness) with our input-side feature sets.

For a given model–task pair, switching the classifier (LogReg vs. MLP) rarely changes outcomes for the baselines, and on the hardest benchmarks, DICE (idiomaticity) and TroFi/MOH-X (metaphor), they often show no signal (F1 = 0 for several large models, e.g., Llama-3.1-8B, Qwen2.5-7B, Qwen2.5-14B). This observation show that coarse confidence summaries are too blunt to capture the context-dependent failures these non-compositional tasks probe.

The sentence-level set (Surprisal + CIS + Entropy + CWS) yields substantial gains on every dataset and for every model family, turning many of the zero-F1 cases into non-trivial detection performance. The gains are especially visible on smaller models (0.5B–1.5B), where errors are more frequent but the input-conditioned belief patterns our features exploit are still systematic.

Classifier choice effects are modest but consistent. The MLP generally provides broader task coverage (fewer near-zero cases) and more stable performance across models and datasets. Logistic regression sometimes edges out MLP on smaller models (≈0.5B–3B), which is consistent with linear separability of these features at lower capacity; the MLP tends to do better on the harder datasets (e.g., TroFi, ConMeC) and on larger models, indicating that our features encode non-trivial structure that benefits from a more expressive classifier.

5.2 Do linguistically localized features improve error prediction?

We next ask whether prediction improves when we localize input-side features to regions marked by the datasets as potential sites of misreading (e.g., the annotated phrase).

As shown in Table 1, span-localized features often provide stronger signal than sentence-level aggregation alone, particularly for larger models.

Tasks	DICE	MOHX	TroFi	PUB 14	ConMeC	DICE	MOHX	TroFi	PUB 14	ConMeC
Baseline	MAX PROBABILITY CONFIDENCE									
Llama-3.1-8B-Instruct	0.0	0.0	0.0	74.8	28.9	0.0	0.0	0.0	68.6	0.0
Llama-3.2-3B-Instruct	60.8	0.0	0.0	95.5	72.2	68.2	0.0	0.0	95.5	72.2
Llama-3.2-1B-Instruct	85.3	76.6	89.2	85.0	98.9	85.0	76.6	89.2	85.0	98.9
Qwen2-1.5B-Instruct	79.5	69.1	74.9	96.0	69.1	76.8	60.9	74.9	96.0	70.4
Qwen2.5-0.5B-Instruct	70.6	63.8	82.7	98.3	89.4	70.8	67.0	82.7	98.3	89.4
Qwen2.5-7B-Instruct-1M	0.0	0.0	1.4	97.2	0.0	0.0	0.0	0.0	97.2	0.0
Qwen2.5-14B-Instruct-1M	0.0	0.0	0.0	87.8	0.0	0.0	0.0	0.0	87.8	0.0
Baseline	LOG PROBABILITY									
Llama-3.1-8B-Instruct	22.8	0.0	0.0	74.8	8.0	0.0	0.0	0.0	74.8	9.5
Llama-3.2-3B-Instruct	74.8	0.0	0.0	95.5	72.2	76.2	0.0	1.1	95.5	72.2
Llama-3.2-1B-Instruct	85.6	76.6	89.2	85.0	98.9	85.3	76.6	89.2	85.0	98.9
Qwen2-1.5B-Instruct	78.8	64.3	74.9	96.0	70.4	73.5	61.5	74.5	96.0	70.4
Qwen2.5-0.5B-Instruct	73.9	66.7	82.7	98.3	89.4	72.9	66.3	82.7	98.3	89.4
Qwen2.5-7B-Instruct-1M	5.6	0.0	0.0	97.2	0.0	0.0	0.0	0.0	97.2	0.0
Qwen2.5-14B-Instruct-1M	0.0	0.0	0.0	87.8	0.0	0.0	0.0	0.0	87.8	0.0
Baseline	ODDBALLNESS									
Llama-3.1-8B-Instruct	13.8	0.0	0.0	73.1	2.2	0.0	0.0	0.0	74.8	0.0
Llama-3.2-3B-Instruct	64.9	0.0	0.0	95.5	72.2	66.0	4.3	2.3	95.5	72.2
Llama-3.2-1B-Instruct	85.2	76.6	89.2	85.0	98.9	85.3	76.6	89.2	85.0	98.9
Qwen2-1.5B-Instruct	77.6	61.0	74.9	96.0	70.4	74.0	64.4	74.8	96.0	70.4
Qwen2.5-0.5B-Instruct	73.6	68.4	82.7	98.3	89.4	72.0	69.4	82.7	98.3	89.4
Qwen2.5-7B-Instruct-1M	2.0	0.0	0.0	97.2	0.0	0.0	0.0	0.0	97.2	0.0
Qwen2.5-14B-Instruct-1M	0.0	0.0	0.0	87.8	0.0	0.0	0.0	0.0	87.8	0.0
Ours (Sentence-level)	SURPRISAL + CIS + ENTROPY + CWS									
Llama-3.1-8B-Instruct	27.7	0.0	0.7	74.3	31.7	41.3	0.0	20.9	66.7	45.9
Llama-3.2-3B-Instruct	74.8	0.0	15.0	95.5	72.0	78.0	27.4	33.3	95.5	61.0
Llama-3.2-1B-Instruct	85.6	74.5	89.1	85.0	98.9	84.8	71.1	89.2	78.7	98.9
Qwen2-1.5B-Instruct	77.4	54.7	71.3	96.0	65.8	73.1	57.1	70.3	96.0	63.3
Qwen2.5-0.5B-Instruct	75.3	67.1	81.9	98.3	89.4	75.5	59.0	81.8	97.8	88.7
Qwen2.5-7B-Instruct-1M	3.7	0.0	1.5	97.2	0.0	27.0	11.4	23.6	97.2	8.4
Qwen2.5-14B-Instruct-1M	0.0	0.0	0.8	87.8	0.0	5.7	9.5	26.1	88.3	33.5
Ours	LINGUISTICALLY TARGETED									
Llama-3.1-8B-Instruct	41.00 (+13.3)	7.14 (+7.1)	18.34 (+17.6)	65.55 (-8.7)	48.02 (+16.4)	50.00 (+8.7)	10.53 (+10.5)	42.07 (+21.1)	65.45 (-1.2)	48.92 (+3.0)
Llama-3.2-3B-Instruct	79.78 (+5.0)	17.24 (+17.2)	40.87 (+25.8)	92.40 (-3.1)	65.79 (-6.2)	76.22 (-1.8)	32.56 (+5.2)	49.40 (+16.1)	94.86 (-0.6)	61.64 (+0.7)
Llama-3.2-1B-Instruct	84.29 (-1.3)	75.86 (+1.4)	89.18 (+0.1)	77.85 (-7.1)	(=)	81.90 (-2.8)	64.20 (-6.9)	84.55 (-4.7)	76.60 (-2.1)	98.80 (-0.1)
Qwen2-1.5B-Instruct	79.30 (+1.9)	66.18 (+11.5)	70.08 (-1.2)	96.00 (-0.0)	59.56 (-6.2)	74.24 (+1.1)	63.45 (+6.3)	70.12 (-0.2)	(=)	57.96 (-5.3)
Qwen2.5-0.5B-Instruct	75.20 (-0.1)	63.69 (-3.4)	80.61 (-1.3)	96.05 (-2.3)	89.31 (-0.1)	71.55 (-3.9)	63.69 (+4.7)	74.79 (-7.0)	98.34 (+0.6)	83.60 (-5.1)
Qwen2.5-7B-Instruct-1M	26.85 (+23.1)	22.86 (+22.9)	9.80 (+8.3)	97.75 (+0.5)	24.55 (+24.5)	42.62 (+15.6)	12.77 (+1.3)	39.55 (+16.0)	97.18 (-0.0)	38.55 (+30.1)
Qwen2.5-14B-Instruct-1M	9.88 (+9.9)	11.11 (+11.1)	13.94 (+13.2)	83.87 (-3.9)	23.97 (+24.0)	27.59 (+21.9)	11.76 (+2.2)	35.20 (+9.1)	84.21 (-4.1)	40.38 (+6.9)

Table 1: Results on logistic regression (left panel) and MLP (right panel) classifiers. Top three panels shows the results obtained using baseline features. Bottom two panels shows performance using features derived from surprisal cues, with and without the presence of linguistic spans. All values presented are F1 scores of detecting error, averaged across three runs.

For example, on Qwen2.5-14B (LogReg), localization yields clear improvements on ConMeC and TroFi, and Qwen2.5-7B shows consistent gains across multiple datasets. This suggests that larger models, while strong on average, benefit from focused measurement at the span and its boundary, where local instability can be more diagnostic than global summaries.

For smaller models (e.g., Llama-3.2-1B, Qwen2.5-0.5B), sentence-level features capture most of the available signal, and span-localization brings only modest, or occasionally negative, changes, consistent with errors that are predominantly global and already reflected in the input-likelihood profile.

Overall, sentence-level features provide a strong, pre-output signal on their own. Span-localized features are complementary, offering clear gains when region of interests are available (or can be inferred) and the base model has capacity for improvement. This points toward future work on automatic localization to bring these benefits without task-specific annotations.

5.3 Where is the information for error?

In this section, we aim to pinpoint the most important signals for error. To this end, we analyze the impact of removing individual surprisal-based cues from the sentence-level models by computing the performance delta across logistic regression and MLP classifiers. As shown in Table 2, the ablations affect the MLP classifier more significantly, which reflects its higher sensitivity to input features and its capacity to model more complex, non-linear interactions than logistic regression.

For logistic regression, most changes are small (often within a few F1 points), with the larger negative deltas appearing when Surprisal or Entropy is removed. In contrast, CIS and CWS generally contribute less under the linear model, with ablations producing little movement in many model–task pairs.

Tasks	DICE	MOHX	TroFi	PUB 14	ConMeC	DICE	MOHX	TroFi	PUB 14	ConMeC
Ablation of CWS	SURPRISAL + CIS + ENTROPY									
Llama-3.1-8B-Instruct	-1.3	0	-0.7	2.0	0.8	-2.2	7.1	-1.8	1.6	-10.4
Llama-3.2-3B-Instruct	-0.4	0	-6.7	0	0.3	0.5	-3.7	3.7	0	4.0
Llama-3.2-1B-Instruct	0.1	-1.2	0.1	0	0	-0.7	-1.7	-0.4	-1.4	0
Qwen2-1.5B-Instruct	-0.6	8.1	3.5	0	3.8	2.5	-3.0	1.9	0	-8.0
Qwen2.5-0.5B-Instruct	-0.5	-3.5	0.8	0	0	-1.5	-13.2	-0.1	0.6	0.3
Qwen2.5-7B-Instruct-1M	3.4	0	3.2	0	0	-14.2	0	0.3	0	10.7
Qwen2.5-14B-Instruct-1M	0	0	-0.0	0	1.0	5.2	0.5	-3.9	-1.4	-8.7
Ablation of Surprisal	CIS + ENTROPY + CWS									
Llama-3.1-8B-Instruct	-0.5	0	-0.7	0.9	0.3	-1.9	7.1	-0.1	0	-4.6
Llama-3.2-3B-Instruct	-3.3	0	-5.3	0	0	-0.0	-4.0	7.2	0	5.8
Llama-3.2-1B-Instruct	-0.9	-1.5	0.1	0	0	-0.5	-2.9	-0.4	-4.9	0
Qwen2-1.5B-Instruct	-0.6	8.1	3.5	0	3.6	1.5	-1.7	2.6	0	-7.4
Qwen2.5-0.5B-Instruct	-0.7	-1.6	0.8	0	0	-2.5	-7.2	0.1	0.6	0.0
Qwen2.5-7B-Instruct-1M	-0.0	0	1.2	0	0	-14.1	0	2.6	0	11.1
Qwen2.5-14B-Instruct-1M	0	0	-0.8	0	1.0	7.4	0.5	-7.4	-2.1	-10.2
Ablation of CIS	SURPRISAL + ENTROPY + CWS									
Llama-3.1-8B-Instruct	-0.6	0	-0.1	-2.5	-2.0	-1.1	0	-12.9	7.2	-15.8
Llama-3.2-3B-Instruct	0.8	0	-2.8	0	0.3	1.6	-6.2	-0.6	0	5.1
Llama-3.2-1B-Instruct	0.0	-1.4	0.1	0	0	-0.3	-2.3	-0.1	2.6	0
Qwen2-1.5B-Instruct	-0.7	9.5	3.4	0	-1.5	1.1	6.0	3.2	0	-3.0
Qwen2.5-0.5B-Instruct	-0.6	1.5	0.8	0	0	-1.0	-5.6	0.2	0.6	0.6
Qwen2.5-7B-Instruct-1M	1.9	0	0.5	0	0	-15.1	-11.4	-3.1	0	6.7
Qwen2.5-14B-Instruct-1M	0	0	-0.8	0	0	-5.7	-9.5	-14.0	-0.7	-28.2
Ablation of Entropy	SURPRISAL + CIS + CWS									
Llama-3.1-8B-Instruct	-0.2	0	-0.7	0.2	-0.2	-1.3	7.4	-5.7	7.0	-16.9
Llama-3.2-3B-Instruct	-0.1	0	-6.9	0	0.1	-0.6	-4.8	-4.5	0	9.1
Llama-3.2-1B-Instruct	-0.3	2.2	0.1	0	0	0.7	1.1	-0.0	0.2	0
Qwen2-1.5B-Instruct	-0.5	3.4	3.7	0	0.3	2.0	-1.1	2.8	0	-2.9
Qwen2.5-0.5B-Instruct	-0.2	-0.4	0.8	0	0	-3.3	-5.0	0.1	0.6	0.5
Qwen2.5-7B-Instruct-1M	1.8	0	-0.1	0	0	-25.0	6.2	-16.7	0	10.3
Qwen2.5-14B-Instruct-1M	0	0	-0.8	0	0	-5.7	9.5	-19.3	-0.5	-20.5

Table 2: Feature ablation results for the logistic regression (left panel) and MLP (right panel) classifiers. Values represent the performance difference (Δ) between the full results on all four metrics and the ablated model results, computed as: Δ = ablated model − full model. All values are results averaged from three runs. Negative values indicate that removing the feature decreases performance (i.e., the feature is important), while Positive values suggest the feature may be redundant or detrimental. Each panel shows the effect of ablating a specific metric.

The pattern shifts with the MLP: CIS and Entropy account for much of the signal, with their ablations producing the largest drops overall, while Surprisal remains useful but is no longer the dominant driver of performance. CWS shows the least consistent impact across both classifiers, since it is essentially surprisal augmented with a confidence/peakiness penalty and is therefore highly collinear with the explicit Surprisal and Entropy features already included, limiting its marginal contribution. Taken together, these results suggest that feature importance depends on the decision surface: a linear model primarily exploits Surprisal/Entropy, whereas a non-linear model leverages interactions that make CIS and Entropy comparatively more informative, with CWS contributing the least.

6 Conclusions

We showed that a language model's *perception of the input*, captured by its token-level probabilities over the given prompt, provides a reliable, *pre-output* signal of error. Operationalizing this idea with simple, interpretable features over the input-side likelihood surface (Surprisal, Entropy, CIS, CWS), our framework anticipates failures without consulting generated outputs or hidden activations, requiring only token log-probabilities. Across five context-sensitive benchmarks and a range of model scales, these structured input-side features consistently recover usable signal where coarse input-likelihood summaries (e.g., mean log likelihood, mean max token probability, Oddballness) often show little or no separability, especially on idiomaticity and metaphor. This establishes a clear result: errors can be foretold from how the model reads the input, not just from what it later says. We further find a consistent deployment pattern in our experiments. Sentence-level features alone deliver strong performance, particularly for smaller models, while *localizing* the measurement to annotated regions of interest yields substantial additional gains for larger models. In practice, this gives a simple recipe: use sentence-level features by default; add span-localized measurement when regions of interest are known or can be inferred. This input-only view is complementary to output- or activation-based detectors and can be composed with them to further improve reliability.

Limitations

While this work focuses on evaluation using English data, this work can be extended to other languages, given availability of evaluation data, as the relevant input signals can be generated and are expected to be similarly informative.

Due to limited research budget, an important direction for future work involves extending our framework to evaluate and leverage cognitively inspired signals from closed-source models such as OpenAI's GPT-4o. These models are increasingly prevalent in deployed NLP systems, yet their opacity poses challenges for extracting internal metrics like layerwise activations or fine-tuned representations.

In our preliminary investigation, we also evaluated several smaller models to assess their performance. Namely, SmolLM models under 2B parameters (Allal et al., 2025). However, we encountered a contradictory challenge: in order to test our classifiers, the LLM models must first be capable of generating meaningful responses to the non-literal language evaluation prompts. Unfortunately, the smaller models consistently failed to produce any correct outputs - they misclassified all instances. In other words, without at least some correct predictions to contrast against the errors, the requirement for error detection of distinguishing between right and wrong responses could not be met.

Acknowledgments

This work was supported by the UKRI AI Centre for Doctoral Training in Speech and Language Technologies (SLT) and their Applications funded by UK Research and Innovation [grant number EP/S023062/1]. For the purpose of open access, the author has applied a Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising. We acknowledge IT Services at The University of Sheffield for the provision of services for High Performance Computing. This research was partly supported by MRC-FAPESP AIM-Health and UniDive COST Action. We are also greateful to members of the Sheffield NLP group for their helpful discussions in this work, in particular, Atsuki Yamaguchi, Ben Wu, Constantinos Karouzos. Finally, we thank the anonymous reviewers and metareviewer for their feedback on this paper.

References

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, Joshua Lochner, Caleb Fahlgren, Xuan-Son Nguyen, Clémentine Fourrier, Ben Burtenshaw, Hugo Larcher, Haojun Zhao, Cyril Zakka, Mathieu Morlon, and 3 others. 2025. Smollm2: When smol goes big – data-centric training of a small language model. *Preprint*, arXiv:2502.02737.

Dhananjay Ashok and Jonathan May. 2025. Language models can predict their own behavior. *arXiv preprint arXiv:2502.13329*.

Amos Azaria and Tom Mitchell. 2023. The internal state of an LLM knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 967–976, Singapore. Association for Computational Linguistics.

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. 2023. Eliciting latent predictions from transformers with the tuned lens. *Preprint*, arXiv:2303.08112.

Julia Birke and Anoop Sarkar. 2006. A clustering approach for nearly unsupervised recognition of nonliteral language. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 329–336, Trento, Italy. Association for Computational Linguistics.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. 2024. Discovering latent knowledge in language models without supervision. *Preprint*, arXiv:2212.03827.

Wallace L Chafe. 1968. Idiomaticity as an anomaly in the chomskyan paradigm. *Foundations of language*, pages 109–127.

Hila Chefer, Shir Gur, and Lior Wolf. 2021. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 782–791.

Yung-Sung Chuang, Linlu Qiu, Cheng-Yu Hsieh, Ranjay Krishna, Yoon Kim, and James R. Glass. 2024. Lookback lens: Detecting and mitigating contextual hallucinations in large language models using only attention maps. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1419–1436, Miami, Florida, USA. Association for Computational Linguistics.

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29.

R.M. Fano. 1961. *Transmission of Information: A Statistical Theory of Communication*. MIT Press Classics. MIT Press.

Bruce Fraser. 1970. Idioms within a transformational grammar. *Foundations of language*, pages 22–42.

Richard Futrell, Edward Gibson, and Roger P. Levy. 2020. Lossy-context surprisal: An information-theoretic model of memory effects in sentence processing. *Cognitive Science*, 44(3):e12814.

Dmitriy Genzel and Eugene Charniak. 2002. Entropy rate constancy in text. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 199–206, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Saptarshi Ghosh and Tianyu Jiang. 2025. ConMeC: A dataset for metonymy resolution with common nouns. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6493–6509, Albuquerque, New Mexico. Association for Computational Linguistics.

Adam Goodkind and Klinton Bicknell. 2018. Predictive power of word surprisal for reading times is a linear function of language model quality. In *Proceedings of the 8th Workshop on Cognitive Modeling and Computational Linguistics (CMCL 2018)*, pages 10–18, Salt Lake City, Utah. Association for Computational Linguistics.

Filip Gralinski, Ryszard Staruch, and Krzysztof Jurkiewicz. 2025. Oddballness: universal anomaly detection with language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 2683–2689, Abu Dhabi, UAE. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Zhengyao Gu and Mark Hopkins. 2023. On the evaluation of neural selective prediction methods for natural language processing. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7888–7899, Toronto, Canada. Association for Computational Linguistics.

John Hale. 2001. A probabilistic Earley parser as a psycholinguistic model. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

Wei He, Marco Idiart, Carolina Scarton, and Aline Villavicencio. 2024. Enhancing idiomatic representation in multiple languages via an adaptive contrastive triplet loss. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12473–12485, Bangkok, Thailand. Association for Computational Linguistics.

T. Jaeger and Roger Levy. 2006. Speakers optimize information density through syntactic reduction. In *Advances in Neural Information Processing Systems*, volume 19. MIT Press.

T Florian Jaeger. 2010. Redundancy and reduction: Speakers manage syntactic information density. *Cognitive psychology*, 61(1):23–62.

Jad Kabbara and Jackie Chi Kit Cheung. 2022. Investigating the performance of transformer-based NLI models on presuppositional inferences. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 779–785, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Roger Levy. 2008. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177.

Roger Levy. 2013. Memory and surprisal in human sentence comprehension. In *Sentence processing.*, Current issues in the psychology of language., pages 78–114. Psychology Press, New York, NY, US.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Inference-time intervention: eliciting truthful answers from a language model. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Huan Ma, Jingdong Chen, Guangyu Wang, and Changqing Zhang. 2025. Estimating llm uncertainty with logits. *arXiv preprint arXiv:2502.00290*.

R Thomas McCoy, Shunyu Yao, Dan Friedman, Mathew D Hardy, and Thomas L Griffiths. 2024. Embers of autoregression show how large language models are shaped by the problem they are trained to solve. *Proceedings of the National Academy of Sciences*, 121(41):e2322420121.

Maggie Mi, Aline Villavicencio, and Nafise Sadat Moosavi. 2025. Rolling the DICE on idiomaticity: How LLMs fail to grasp context. In *Proceedings* *of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7314–7332, Vienna, Austria. Association for Computational Linguistics.

Saif Mohammad, Ekaterina Shutova, and Peter Turney. 2016. Metaphor as a medium for emotion: An empirical study. In *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*, pages 23–33, Berlin, Germany. Association for Computational Linguistics.

Byung-Doh Oh and William Schuler. 2023. Transformer-based language model surprisal predicts human reading times best with about two billion training tokens. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1915–1921, Singapore. Association for Computational Linguistics.

Masanari Ohi, Masahiro Kaneko, Ryuto Koike, Mengsay Loem, and Naoaki Okazaki. 2024. Likelihood-based mitigation of evaluation bias in large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3237–3245, Bangkok, Thailand. Association for Computational Linguistics.

Andreas Opedal, Eleanor Chodroff, Ryan Cotterell, and Ethan Wilcox. 2024. On the role of context in reading time prediction. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3042–3058, Miami, Florida, USA. Association for Computational Linguistics.

Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. 2017. Regularizing neural networks by penalizing confident output distributions. *Preprint*, arXiv:1701.06548.

Dylan Phelps, Thomas Pickard, Maggie Mi, Edward Gow-Smith, and Aline Villavicencio. 2024. Sign of the times: Evaluating the use of large language models for idiomaticity detection. In *Proceedings of the Joint Workshop on Multiword Expressions and Universal Dependencies (MWE-UD) @ LREC-COLING 2024*, pages 178–187, Torino, Italia. ELRA and ICCL.

Tiago Pimentel, Clara Meister, Ethan G. Wilcox, Roger P. Levy, and Ryan Cotterell. 2023. On the effect of anticipation on reading times. *Transactions of the Association for Computational Linguistics*, 11:1624–1642.

Ernesto Quevedo, Jorge Yero Salazar, Rachel Koerner, Pablo Rivas, and Tomas Cerny. 2025. Detecting hallucinations in large language model generation: A token probability approach. In *Artificial Intelligence and Applications*, pages 154–173, Cham. Springer Nature Switzerland.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Giulia Rambelli, Emmanuele Chersoni, Marco S. G. Senaldi, Philippe Blache, and Alessandro Lenci. 2023. Are frequent phrases directly retrieved like idioms? an investigation with self-paced reading and language models. In *Proceedings of the 19th Workshop on Multiword Expressions (MWE 2023)*, pages 87–98, Dubrovnik, Croatia. Association for Computational Linguistics.

C. E. Shannon. 1948. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423.

Ola Shorinwa, Zhiting Mei, Justin Lidard, Allen Z Ren, and Anirudha Majumdar. 2024. A survey on uncertainty quantification of large language models: Taxonomy, open research challenges, and future directions. *arXiv preprint arXiv:2412.05563*.

Nathaniel J Smith and Roger Levy. 2013. The effect of word predictability on reading time is logarithmic. *Cognition*, 128(3):302–319.

Settaluri Sravanthi, Meet Doshi, Pavan Tankala, Rudra Murthy, Raj Dabre, and Pushpak Bhattacharyya. 2024. PUB: A pragmatics understanding benchmark for assessing LLMs' pragmatics capabilities. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12075–12097, Bangkok, Thailand. Association for Computational Linguistics.

Gerard J. Steen, Aletta G. Dorst, J. Berenike Herrmann, Anna A. Kaal, Tina Krennmayr, and Tryntje Pasma. 2010. *A Method for Linguistic Metaphor Identification*. John Benjamins.

Yuan Tian, Ruike Zhang, Nan Xu, and Wenji Mao. 2024. Bridging word-pair and token-level metaphor detection with explainable domain mining. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13311–13325, Bangkok, Thailand. Association for Computational Linguistics.

Eleftheria Tsipidi, Franz Nowak, Ryan Cotterell, Ethan Wilcox, Mario Giulianelli, and Alex Warstadt. 2024. Surprise! Uniform Information Density isn't the whole story: Predicting surprisal contours in long-form discourse. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18820–18836, Miami, Florida, USA. Association for Computational Linguistics.

Jesse Vig and Yonatan Belinkov. 2019. Analyzing the structure of attention in a transformer language model. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 63–76, Florence, Italy. Association for Computational Linguistics.

Ethan G. Wilcox, Tiago Pimentel, Clara Meister, Ryan Cotterell, and Roger P. Levy. 2023. Testing the predictions of surprisal theory in 11 languages. *Transactions of the Association for Computational Linguistics*, 11:1451–1470.

Yorick Wilks. 1975. A preferential, pattern-seeking, semantics for natural language inference. *Artificial Intelligence*, 6(1):53–74.

Yorick Wilks. 1978. Making preferences more active. *Artificial Intelligence*, 11(3):197–223.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Huggingface’s transformers: State-of-the-art natural language processing. *Preprint*, arXiv:1910.03771.

Jialiang Wu, Yi Shen, Sijia Liu, Yi Tang, Sen Song, Xiaoyi Wang, and Longjun Cai. 2025. Improve decoding factuality by token-wise cross layer entropy of large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 3912–3921, Albuquerque, New Mexico. Association for Computational Linguistics.

Yang Xu and David Reitter. 2016. Entropy converges between dialogue participants: Explanations from an information-theoretic perspective. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 537–546, Berlin, Germany. Association for Computational Linguistics.

Cheng Yang, Puli Chen, and Qingbao Huang. 2024. Can ChatGPT‘s performance be improved on verb metaphor detection tasks? bootstrapping and combining tacit knowledge. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1016–1027, Bangkok, Thailand. Association for Computational Linguistics.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, and 2 others. 2025. Representation engineering: A top-down approach to ai transparency. *Preprint*, arXiv:2310.01405.

A Additional Linguistics Features

Context-level Span We define context-level span as considering the surprisal distribution, surround the targeted expression.

$$k_{\text{ctx}}(x) = k_{\text{left}}(x) \cup k_{\text{right}}(x),$$

where:

$$k_{\text{left}}(x) \subset \{1, \ldots, \min(k_{\text{expr}}(x)) - 1\},$$
$$k_{\text{right}}(x) \subset \{\max(k_{\text{expr}}(x)) + 1, \ldots, n\}.$$

Relative magnitude of max surprisal in the idiom To assess the relative importance of surprisal within the idiom, we define the following ratio. Let:

$$s_{\max}^{\text{expr}} = \max_{i \in S_{\text{expr}}} k(x_i), \quad S_{\text{rest}} = \sum_{i \in j \setminus S_{\text{expr}}} k(x_i), \tag{6}$$

$$R_{\text{contrast}} = \frac{s_{\max}^{\text{expr}}}{S_{\text{rest}} + \varepsilon}, \tag{7}$$

where $\varepsilon > 0$ is a small constant added for numerical stability. This feature quantifies how prominent the idiom’s most surprising token is relative to the rest of the sentence.

Position of min/max We find the position of the smallest and largest token in a sentence. Then, normalize this position by the total number of tokens in the sentence.

$$i_{\min} = \arg \min_{1 \le j \le n} k(t_j), \quad i_{\max} = \arg \max_{1 \le j \le n} k(t_j),$$

$$p_{\min} = \frac{i_{\min}}{n}, \quad p_{\max} = \frac{i_{\max}}{n},$$

where $k(t_j)$ is the scoring function applied to token t_j.

B Experimental Setup

B.1 Prompts

DICE "Is the expression ’{target_expression}’ used figuratively or literally in the sentence: {sentence} Answer ’i’ for figurative, ’l’ for literal. Put your answer after ’output: ’."

MOH-X and TroFi "Is the word ’{target_word}’ used metaphorically or literally in the sentence: sentence Answer ’m’ for metaphorical, ’l’ for literal. Put your answer after ’output: ’."

PUB 14 Metonymy The PUB Task 14 dataset provides existing task instructions in this format:

> **Context:** She is attracted to blue jacket.
> **Question:** What does ‘blue jacket’ refer to?
> **Choices:**
>
> $$‘Colour', ‘Jacket', ‘Sailor', ‘Sea'$$

Tasks	DICE	MOHX	TroFi	PUB 14	ConMeC	DICE	MOHX	TroFi	PUB 14	ConMeC
Llama-3.1-8B-Instruct	20.6	0.0	0.0	74.5	28.9	5.7	0.0	2.6	66.7	31.1
Llama-3.2-3B-Instruct	74.5	0.0	0.6	95.5	72.4	78.7	15.6	20.3	95.5	67.1
Llama-3.2-1B-Instruct	85.2	76.6	89.2	85.0	98.9	85.0	72.7	89.2	84.3	98.9
Qwen2-1.5B-Instruct	79.7	63.5	74.9	96.0	68.7	76.5	59.7	74.0	96.0	67.6
Qwen2.5-0.5B-Instruct	73.6	65.2	82.7	98.3	89.4	73.7	60.6	82.8	98.3	89.4
Qwen2.5-7B-Instruct-1M	3.7	0.0	3.3	97.2	0.0	0.0	0.0	2.7	97.2	0.0
Qwen2.5-14B-Instruct-1M	0.0	0.0	0.0	87.8	0.0	0.0	0.0	1.5	85.7	1.9

Table 3: Results on logistic regression (left panel) and MLP (right panel) classifiers, trained on all three baselines. All values presented are F1 scores of detecting error, averaged across three runs.

Therefore, we formulate our prompt as:

> The following are multiple choice questions.
> `[pretext]`
> Your options are:
> `[choices]`

ConMeC "Is the word '{target_word}' used metonymically or literally in the sentence: {sentence} Answer 'm' for metonymical, 'l' for literal. Put your answer after 'output: '."

B.2 Model Access

We use HuggingFace (Wolf et al., 2020) to evaluate: Llama 3 models (Grattafiori et al., 2024) and Qwen 2.5 models (Qwen et al., 2025).

We use two A100 GPUs to complete all our feature extraction aspects, as these require (1) prompting LLMs and (2) obtaining the logits from LLMs.

The classifier part of our work can be run on CPUs.

B.3 Classifiers

B.3.1 MLP

We employ a multi-layer perceptron to perform binary classification over $\mathcal{F}$. The architecture consists of three fully connected layers: an input layer mapping to 512 hidden units, a second hidden layer of 512 units, and a final output layer of a single unit. The two hidden layers are followed by ReLU activation functions, and the output layer employs a sigmoid activation to produce a scalar probability. This design follows that used in (Quevedo et al., 2025).

Formally, given an input vector $x \in \mathbb{R}^d$, the output y is computed as:

$$\begin{aligned} h_1 &= \text{ReLU}(W_1 x + b_1), \\ h_2 &= \text{ReLU}(W_2 h_1 + b_2), \\ y &= \sigma(W_3 h_2 + b_3), \end{aligned}$$

where σ denotes the sigmoid function.

Prior to training, all features are standardized to zero mean and unit variance.

We partition the dataset into training and validation splits using an 80/20 ratio, stratified to preserve class distribution across splits. This stratification is critical given the class imbalance commonly observed in error detection tasks.

Training is conducted using the binary cross-entropy loss, optimized via the Adam optimizer with a fixed learning rate of 10^{-3}. Models are trained for 20 epochs with mini-batches of size 32. All computations are performed using PyTorch. At each epoch, we monitor both the binary cross-entropy loss and classification accuracy on the training set. Model evaluation is deferred until training completion.

B.3.2 Logistic Regression

We employ a regularized **logistic regression** classifier to perform binary classification over $\mathbf{E} \in \{0, 1\}$. Given an input vector $\mathcal{F}$, the predicted probability y is given by:

$$y = \sigma(w^\top x + b),$$

where $w \in \mathbb{R}^d$ is the weight vector, b is the bias term, and $\sigma(\cdot)$ denotes the sigmoid function.

We train the model using the `lbfgs` solver with an increased maximum number of iterations (2,500) to ensure convergence on the standardized feature set. For each task, the dataset is split into training (80%) and testing (20%) sets using stratified sampling to maintain class balance across splits.

To ensure compatibility with linear models and to improve convergence, all features are standardized using `StandardScaler` to have zero mean and unit variance.

B.4 Evaluation Metrics

The performance of the classifiers is evaluated using the predicted labels on the held-out test set. We compute standard classification metrics including **precision**, **recall**, and **F1 score**, as well as class-wise scores. These metrics provide insights into the classifier's ability to correctly identify both positive (error) and negative (non-error) instances.

B.5 Baseline Definitions

Log Likelihood In line with (McCoy et al., 2024), we compute log-likelihood of a text sequence using a causal language model.

$$\frac{1}{n}\sum_{i=1}^{n}\log P(t_i \mid t_1, t_2, \ldots, t_{i-1})$$

Max Token Probability This function computes a confidence score for a given text based on a language model's predictions. For each token in the input (except the last, due to causal shifting), it calculates the maximum probability assigned to any token in the vocabulary at that position, which represents the model's confidence in its top prediction. These maximum probabilities are then aggregated using mean across the sequence using a specified method to produce a single scalar confidence value. This score reflects how confident the model is, on average or otherwise, about its predictions across the entire input sequence.

$$\text{Confidence} = \frac{1}{n-1}\sum_{i=1}^{n-1}\max_{v\in V} P(v \mid t_1, \ldots, t_i)$$

Oddballness We calculate the maximum Oddballness (Gralinski et al., 2025) of a text sequence using a decoder-only language model. For each token in the sequence, it computes the softmax distribution over the vocabulary and compares the probability assigned to the actual token with all other tokens. The difference between these values (only when positive) is passed through a ReLU, summing the total surplus probability assigned to alternative tokens. This gives a per-token oddballness score, reflecting how much more likely the model thought other tokens were compared to the actual one. The function returns the maximum of these scores across the sequence, indicating the point where the model found the actual token most surprising or inconsistent.

$$\text{oddball}_i = \sum_{v\in V}\text{ReLU}(P_i[v] - p_i)$$

C Complementary Results

C.1 Model Performance on All Tasks

Table 4 presents the performance of the evaluated LLMs on all five non-literal language tasks.

Model	DICE	MOHX	TroFi	PUB 14	ConMeC
Llama-3.1-8B-Instruct	69.8	80.7	62.8	40.2	54.0
Llama-3.2-3B-Instruct	56.1	66.8	56.2	8.7	43.7
Llama-3.2-1B-Instruct	25.7	38.3	20.9	26.2	2.2
Qwen2-1.5B-Instruct	34.7	48.8	42.8	7.2	45.6
Qwen2.5-0.5B-Instruct	43.5	45.7	31.2	3.1	18.9
Qwen2.5-7B-Instruct-1M	76.3	77.1	63.5	5.5	62.4
Qwen2.5-14B-Instruct-1M	83.7	86.1	66.9	21.4	62.3

Table 4: Models performance results (classification accuracy) on each task.

C.2 Combined Baselines

We further evaluate a model using all three baseline feature sets jointly. The same training and evaluation protocols as in the main experiments are applied, except that the feature vectors are concatenated before training the classifiers. Results are presented in Table 3.