Word Surprisal Correlates with Sentential Contradiction in LLMs

Ning Shi **Bradley Hauer** **David Basil** **John Zhang** **Grzegorz Kondrak**
Alberta Machine Intelligence Institute (Amii)
Department of Computing Science
University of Alberta, Edmonton, Canada
{ning.shi,gkondrak}@ualberta.ca

Abstract

Large language models (LLMs) continue to achieve impressive performance on reasoning benchmarks, yet it remains unclear how their predictions capture semantic consistency between sentences. We investigate the important open question of whether word-level surprisal (a measure of the estimated unpredictability of a word) correlates with sentence-level contradiction between a premise and a hypothesis. Specifically, we compute surprisal for hypothesis words across a diverse set of experimental variants, and analyze its association with contradiction labels over multiple datasets and open-source LLMs. Because modern LLMs operate on subword tokens and can not directly produce reliable surprisal estimates, we introduce a token-to-word decoding algorithm that extends theoretically grounded probability estimation to open-vocabulary settings. Experiments show a consistent and statistically significant positive correlation between surprisal and contradiction across models and domains. Our analysis also provides new insights into the capabilities and limitations of current LLMs. Together, our findings suggest that surprisal is associated with sentence-level inconsistency at the word level, establishing a quantitative link between lexical uncertainty and sentential semantics.

1 Introduction

Large language models (LLMs) have achieved remarkable success in diverse language understanding and generation benchmarks (Suzgun et al., 2023). This progress has motivated growing interest in understanding how language models capture reasoning, and in providing transparency to end users (Huang and Chang, 2023). This is particularly true for semantic tasks such as recognizing textual entailment (Dagan et al., 2005) and natural language inference (NLI; MacCartney, 2009). However, analysis of LLM outputs is complicated by the fact that existing models are primarily optimized for task-specific performance (Tedeschi et al., 2023), and often lack well-defined objectives or linguistic grounding (Nityasya et al., 2023). Prior work has examined semantic relations at the word and sentence levels (Shi et al., 2024a,b), motivating our study to bridge the two. In particular, a notable scarcity of research examining the connection between lexical and sentential semantics (Hauer and Kondrak, 2023; Wu et al., 2023) leaves unresolved an important open question of how the lexical meaning of words contributes to an LLM's ability to capture sentential understanding and reasoning. In particular, how does a language model respond to semantic inconsistency between two given sentences?

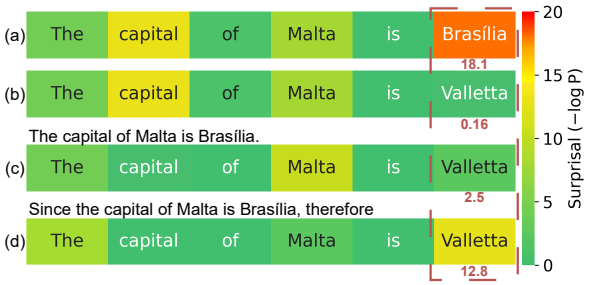

Figure 1: Word-level surprisal heatmaps illustrating how model expectations shift with and without contextual premises. Color intensity indicates surprisal for each word in the hypothesis. In (a) and (b), no premise is provided: (a) shows high surprisal for the false fact ("*Brasília*"), while (b) shows low surprisal for the true fact ("*Valletta*"). In (c) and (d), the model is conditioned on an objectively false premise: (c) shows increased surprisal for the true fact, which further rises dramatically in (d) when the premise is emphasized causally.

In this paper, we examine whether word surprisal correlates with semantic contradiction between two sentences. We posit that there exists a positive correlation between high surprisal of content words in a *hypothesis* sentence, and the existence of contradiction with respect to its *premise*.

This idea is motivated by the synergistic goals of (a) improving understanding of the reasoning capabilities of LLMs, (b) increasing the explainability of LLM outputs, and (c) bridging the gap between lexical and sentential semantics. By testing this correlation empirically, we aim to establish a quantifiable relationship that connects sentence-level contradiction to word-level surprisal.

The computation of word surprisal in LLMs is complicated by the fact that modern models predict probabilities over subword tokens rather than full words. As a result, computing surprisal directly from token probabilities can produce distorted estimates due to boundary ambiguity and cumulative probability over- or underestimation (Oh and Schuler, 2024; Pimentel and Meister, 2024). To address this issue, we introduce a *token-to-word decoding* algorithm that extends theoretically grounded word probability estimation to open-vocabulary settings. The algorithm integrates constrained beam search, dynamic normalization, and end-of-word (EOW) simulation, to achieve accurate and efficient surprisal estimation. It enables scalable computation of word surprisal in token-based LLMs, which provides the computational foundation for our empirical correlation analysis.

We conduct an empirical analysis across multiple datasets and widely-used open-source LLMs. The results demonstrate a statistically significant positive correlation between surprisal and contradiction. We further show that surprisal-based classifiers achieve performance significantly above random chance in contradiction detection. Extended analyses reveal that structured prompts (e.g., causal templates) substantially improve model reliability, even when the premise is objectively false. Finally, our findings also emphasize the persistent challenges that LLMs face, such as maintaining logical coherence and understanding negation, indicating that they still struggle with basic forms of sentential reasoning. We make our code and data available on GitHub.

Our contributions are as follows:

1. We present the first systematic correlation analysis between lexical surprisal and sentential contradiction in LLMs.

2. We introduce a token-to-word decoding algorithm for accurate word-level surprisal estimation from token probabilities in open-vocabulary settings.

3. We provide empirical evidence across datasets and model families showing that word-level surprisal positively correlates with sentence-level contradiction.

2 Theoretical Formulation

This section establishes the theoretical foundation of our study. We adopt the standard definition of contradiction from NLI, in which contradiction is defined as a semantic relation between a premise and a hypothesis. We also adopt the standard notion of surprisal as a probabilistic measure of the unexpectedness of a word in context. Treating surprisal as a continuous measure and contradiction as a binary relation allows us to frame their connection as a problem of measuring correlation, providing the conceptual bridge that motivates our research.

2.1 Natural Language Inference

NLI concerns the logical relationship between a premise P and a hypothesis H. It has become a central benchmark for evaluating natural language understanding (Bowman et al., 2015; Williams et al., 2018; Nie et al., 2020), as it requires models to move beyond surface patterns and assess semantic consistency between sentences. NLI is typically modeled in terms of three semantic relations (MacCartney, 2009). Given a premise P, a hypothesis H may be classified as *true* (entailment), *false* (contradiction), or *undetermined* (neutral). Among these relations, contradiction is as fundamental as entailment (Katz, 1972; van Benthem, 2008), although prior work has often emphasized the latter (Dagan et al., 2010). Recognizing contradiction has been argued to constitute a minimal criterion for text understanding: a system that fails to identify contradictory sentences can not be said to fully understand them (Condoravdi et al., 2003).

2.2 Contradiction Detection

While there are multiple ways to define contradiction (Harabagiu et al., 2006; de Marneffe et al., 2008; Ritter et al., 2008), we follow the standard NLI formulation (MacCartney and Manning, 2009). We focus on the binary relation of contradiction, making no distinction between entailment and neutrality. Formally, a contradiction (CON) is defined as a relation between a premise P and a hypothesis H such that P entails the negation of H:

$$\text{CON}(P, H) \;\Leftrightarrow\; P \models \neg H$$

Contradiction presents a more challenging task than entailment (de Marneffe et al., 2008). While

entailment can sometimes be approximated by surface overlap or similarity in word usage, contradiction requires identifying genuine semantic opposition and logical inconsistency. Accurate modeling of contradiction is therefore critical both for theoretical accounts of semantic relations, and for practical applications such as fact-checking (Nie et al., 2019).

2.3 Surprisal

Surprisal measures the unpredictability of a word in context, originally defined in information theory as the informational content of an event (Shannon, 1948). Formally, following prior work in psycholinguistics (Hale, 2001), the surprisal of a word w given a preceding context C is defined as the negative log of its conditional probability:

$$\text{Surprisal}(w \mid C) = -\log \mathcal{P}(w \mid C) \quad (1)$$

Surprisal quantifies how unexpected a word is in context: the lower its probability, the higher its surprisal. In psycholinguistics, it is interpreted as the cognitive cost of disconfirming competing structural or lexical predictions. It can also be formalized as the relative entropy between prior and updated parse distributions (Levy, 2008). Empirical studies confirm that surprisal reliably correlates with human reading behavior (Smith and Levy, 2013; Wilcox et al., 2023).

In computational settings, surprisal is often estimated from the next-word probability distribution of a language model. Early work used n-gram models, followed by recurrent neural networks such as LSTMs. Current approaches rely on Transformer-based causal language models (CLMs), which directly provide next-token probabilities (Goodkind and Bicknell, 2018; Radford et al., 2018). However, because these models operate at the subword token level, additional processing is needed to obtain accurate word-level surprisal estimates (Oh and Schuler, 2024; Pimentel and Meister, 2024).

2.4 Correlation Hypothesis

From this foundation, we posit that:

> *When H is conditioned on P, the surprisal magnitude of content words in H is positively correlated with the presence of a contradiction, CON(P, H).*

This hypothesis bridges lexical and sentential semantics by grounding sentence-level contradiction in word surprisal.

Two lines of work motivate this hypothesis. In computational semantics, language models have been shown to capture relations such as entailment (Merrill et al., 2022, 2024). In psycholinguistics, surprisal is closely tied to human reading effort (Hale, 2001; Levy, 2008; Smith and Levy, 2013; Wilcox et al., 2023). Less predictable words slow readers down, showing that surprisal connects word statistics with semantic interpretation. However, most existing studies remain at the sentence level, focus on entailment rather than contradiction, and do not directly link surprisal to semantics.

3 Method

We now describe how surprisal is operationalized to test its correlation with the presence of contradiction. The method varies along three dimensions: how the premise-hypothesis pair is presented, how surprisal is computed, and how multiple surprisal values over the hypothesis are aggregated.

3.1 Context Construction

The way a pair of premise and hypothesis (P, H) is presented to the model determines the context in which surprisal is computed. We experiment with three constructions, illustrated in Figure 1.

H-only Only the hypothesis is provided to the model, testing its internal expectations without any additional information from the premise. This setting examines how much world knowledge the model can apply on its own, a challenge noted in prior work (Ritter et al., 2008).

CAT The premise and hypothesis are concatenated, with the premise directly followed by the hypothesis. This allows surprisal in H to be computed while conditioning on information from P.

TEMP The premise and hypothesis are embedded in a causal frame, for example, "*Since* $\{P\}$, *therefore* $\{H\}$". This structure introduces explicit relational cues, encouraging the model to interpret P as the cause of H.

3.2 Surprisal Computation

Given a left-to-right prefix context C, a CLM produces a probability distribution over the vocabulary $\mathcal{V}$, modeling $\mathcal{P}(w|C)$ $\forall w \in \mathcal{V}$, the likelihood of each word immediately following the given context (Radford et al., 2018). These probabilities serve as the basis for computing surprisal. We consider two variants of this computation.

Direct The raw surprisal value of a word is taken directly as defined in Eq. 1.

Relative The surprisal of a word is compared to that of the most likely next word: $\delta(w') = \log \max_{w \in V} P(w \mid C) - \log P(w' \mid C)$. This adjustment is designed to normalize for contexts where all next-word probabilities are low, emphasizing the model's relative preference rather than absolute uncertainty. Both variants yield word-level surprisal values through our token-to-word decoding algorithm (Section 4) which are subsequently aggregated into a single signal for correlation analysis.

3.3 Surprisal Aggregation

Since a hypothesis H consists of multiple words, we represent its word-level surprisal values as a single signal for correlation analysis. We consider the following variants: [1]

- **Last**: the surprisal of the final word.
- **Max**: the highest surprisal among the words.
- **Mean**: the average surprisal across the words.

These variants reflect different assumptions about how contradiction correlates with surprisal. By consolidating multiple word-level surprisal values into one, each aggregation method can be paired with a decision threshold, thereby reducing the sentence-level task of contradiction detection to word-level surprisal estimation (Hauer and Kondrak, 2022). In particular, `Last` is motivated by evidence that transformer models (Vaswani et al., 2017) often centralize computation at the last token (Mamidanna et al., 2025).

4 Token-to-Word Decoding

To compute and aggregate surprisal, we need reliable word-level probabilities, as defined in Eq. 1. Modern CLMs, however, generate probabilities for subword tokens, where word boundaries are often ambiguous. Previous studies have shown that token-level estimates can distort word-level surprisal, thus producing misleading results (Oh and Schuler, 2024). To address this, we reformulate how word probabilities are derived from token distributions, and propose a token-to-word decoding algorithm for computing accurate word-level probabilities. It builds on established theoretical formulations (Pimentel and Meister, 2024) and further extends them to the open-vocabulary setting.

[1]Note that "words" here refers to content words.

4.1 Definitions

Given a word $w \in \mathcal{V}$ and its preceding context C, the goal is to compute the probability distribution $\mathcal{P}(w \mid C)$ over the word space $\mathcal{V}$. Since modern language models operate over tokens, the word probability is estimated as the probability of generating its token sequence $\mathbf{S}^w$ conditioned on the tokenized context $\mathbf{S}^C$:

$$\mathcal{P}(w \mid C) = \mathcal{P}(\mathbf{S}^w \mid \mathbf{S}^C) \tag{2}$$

Given $\mathbf{S}^w = (s_1, \ldots, s_n)$ for $s_i \in \mathcal{S}$, the conventional approach computes this using the autoregressive factorization of the language model:

$$\mathcal{P}(\mathbf{S}^w \mid \mathbf{S}^C) = \prod_{i=1}^{n} \mathcal{P}(s_i \mid \mathbf{S}^C, s_1, \ldots, s_{i-1}) \tag{3}$$

Although this computation is intuitive and works for models that use EOW marking tokenizers, it does not directly apply to those that use beginning-of-word (BOW) marking schemes (e.g., Llama, and all other LLMs used in this paper). In these models, tokens that begin a new word are explicitly marked, but no signal indicates that a word has ended.

For example, using "_" as the BOW symbol, the token "*_John*" may represent the full word "*John*", but it can also serve as the prefix of longer words like "*Johnathan*" or "*Johnson*". Since the model does not know whether to stop or continue the current word, the estimated probability of the token "*_John*" includes mass that actually belongs to longer continuations, leading to overestimation.

Additional problems that arise when computing word probabilities from token-level outputs include: (1) probabilities are normalized over the token space, making them incompatible with word-level reasoning; (2) the total probability assigned to all word candidates can exceed one, violating the Kolmogorov axiom (Kolmogorov, 1933); and (3) surprisal values may be misaligned due to boundary uncertainty, among others.

4.2 Normalization-Based Correction

To resolve these issues, Pimentel and Meister (2024) propose a normalization-based correction, referred to as a "bug" fix. The corrected form of the word probability in Eq. 2 is given by:

$$\underbrace{\mathcal{P}(w \mid C) = \mathcal{P}(\mathbf{s}^w \mid \mathbf{s}^C)}_{Eq.\ 2} \underbrace{\frac{\sum_{s \in \mathcal{S}_{\text{bow}}} \mathcal{P}(s \mid \mathbf{s}^C, \mathbf{s}^w)}{\sum_{s \in \mathcal{S}_{\text{bow}}} \mathcal{P}(s \mid \mathbf{s}^C)}}_{\text{"bug" fix}} \tag{4}$$

Here, $\mathcal{S}_{\text{bow}} \subset \mathcal{S}$ denotes the set of tokens that indicate the beginning of a word, typically marked by a dedicated leading character (e.g., "_" or "Ġ"). The numerator estimates the probability that the model stops generating after w, while the denominator adjusts for how likely the model was to begin a new word given the context alone.

While the correction in Eq. 4 provides a theoretically grounded adjustment for word probability estimation, it has practical limitations. First, it assumes that the word w is given, and thus does not support open-vocabulary inference where one must search over the word space. Second, it requires access to the full next-token distribution at multiple positions, making it prohibitively expensive when applied repeatedly across candidate words. Finally, it does not constrain or renormalize the token space based on decoding state, which can lead to overestimation or invalid word formations. These limitations make the precise correction infeasible in settings that require efficient word-level probability estimation over an open vocabulary, such as ranking or scoring candidate words.

4.3 Algorithm

In this section, we introduce Token-to-Word Decoding, a constrained beam search algorithm for estimating word probabilities over token space in open-vocabulary settings, as shown in Algorithm 1.

Dynamic Normalization At each decoding step, the token space is dynamically filtered based on the current beam search depth d. This is implemented through the function $\text{Normalize}(\mathcal{P}, \mathcal{S}')$, which operates in two stages. First, it masks all tokens in $\mathcal{P}$ that are not in the provided token set $\mathcal{S}' \subset \mathcal{S}$ by setting their probabilities to zero. Second, it renormalizes the remaining values to ensure that $\mathcal{P}$ forms a valid probability distribution. The choice of $\mathcal{S}'$ depends on the decoding depth:

$$\mathcal{S}' = \begin{cases} \mathcal{S}_{\text{bow}} & \text{if } d = 1 \\ \mathcal{S}_{\text{mid}} & \text{if } d > 1 \end{cases} \tag{5}$$

where $\mathcal{S}_{\text{mid}} = \mathcal{S} \setminus \mathcal{S}_{\text{bow}}$.[2] This dynamic normalization enforces structural consistency while preserving compatibility with the autoregressive factorization. Notably, the normalization at $d = 1$ corresponds to the denominator of the correction term in Eq. 4. The renormalization over mid-word tokens at $d > 1$ constrains decoding to remain within the valid word space. This does not lead to infinite generation, as a virtual EOW event is injected at each step to enable termination, as described below.

[2] In practice, the exact $\mathcal{S}_{\text{bow}}$ and $\mathcal{S}_{\text{mid}}$ depends on the tokenizer and the language. For example, some words include non-alphabetical characters (e.g., "*N'Djamena*").

Algorithm 1 Token-to-Word Decoding

Input: tokenized context $\mathbf{S}^C$, beam width W, beam depth D, language model $\text{F}(\cdot)$

$B_0 \leftarrow \{\langle 0, \mathbf{S}^C \rangle\}$
for $d \in \{1, \ldots, D\}$:
  $B \leftarrow \varnothing$
  for $\langle p, \mathbf{S} \rangle \in B_{d-1}$:
    if $\mathbf{S}.\text{last}() = \textit{EOW}$:
      $B.\text{add}(\langle p, \mathbf{S} \rangle)$; **continue**
    $\mathcal{P} \leftarrow \text{F}(\mathbf{S})$
    if $d = 1$:
      $\mathcal{P} \leftarrow \text{Normalize}(P, \mathcal{S}_{\text{bow}})$ ▷ Eq. 5
    else:
      $\mathcal{P} \leftarrow \text{InjectEOW}(\mathcal{P}, \mathcal{S}_{\text{bow}})$ ▷ Eq. 6
      $\mathcal{P} \leftarrow \text{Normalize}(\mathcal{P}, \mathcal{S}_{\text{mid}})$ ▷ Eq. 5
    for $(p', s) \in \text{Top}(\mathcal{P}, \text{W})$:
      $B.\text{add}(\langle p \cdot p', \mathbf{S} \circ s \rangle)$
  $B_d \leftarrow B.\text{top}(\text{W})$
return $B_D.\text{sort}()$

EOW Injection To enable termination, we inject a virtual EOW event at each step after the first. Since there are no explicit EOW tokens, we simulate termination by reallocating the probability mass of BOW tokens. The underlying assumption is that the probability of starting a new word is equivalent to the probability of ending the current one. Specifically, we sum the total mass assigned to $\mathcal{S}_{\text{bow}}$ and reassign it to a virtual token $\texttt{EOW} \in \mathcal{S}_{\text{mid}}$, defined as:

$$\mathcal{P}(\textit{EOW} \mid \mathbf{s}^C, \mathbf{s}^w) = \textstyle\sum_{s \in \mathcal{S}_{\text{bow}}} \mathcal{P}(s \mid \mathbf{s}^C, \mathbf{s}^w) \tag{6}$$

This operation is performed by the function $\text{InjectEOW}(\mathcal{P}, \mathcal{S}_{\text{bow}})$, applied before normalization. During decoding, generating the EOW token indicates the end of the token path and marks the token sequence as a complete word. This corresponds to the numerator of the correction term in Eq. 4 for the probability of termination following a given token sequence.

Following the example introduced by Oh and Schuler (2024), our algorithm resolves the issue of incorrect allocation of word-by-word surprisal as shown in Figure 2. Instead of searching all token sequences of depth D, which has complexity $O(|\mathcal{S}|^D)$, it keeps only the top beam width candidates, restricts the token space, and allows early termination. This yields a practical complexity

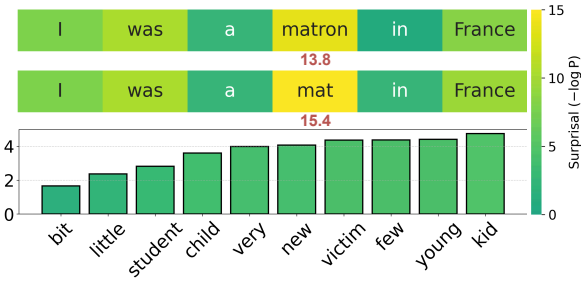

Figure 2: Surprisal values for the example proposed by Oh and Schuler (2024). This case is challenging for token-level methods, but our algorithm correctly assigns higher surprisal to "*mat*" (15.4) than to "*matron*" (13.8). Moreover, it supports open-vocabulary retrieval from the next-word distribution via token-to-word decoding.

of $O(D \cdot |\mathcal{S}'| \cdot W)$, where $\mathcal{S}' \ll \mathcal{S}$. Overall, our algorithm estimates word probabilities through constrained beam search, combining two key components: dynamic normalization and EOW injection. It enables accurate token-to-word decoding at scale, while avoiding the exponential cost of full token path enumeration.

5 Experiments

Our experiments are designed to evaluate the correlation hypothesis (Section 2.4) by focusing on the relationship between word-level surprisal and sentence-level contradiction. The aim is to provide statistical evidence for our claims.

5.1 Data

We conduct experiments on three datasets derived from existing benchmarks. Since contradiction has received relatively little direct attention, dedicated benchmark resources are limited. For this study, we adapt three existing datasets into balanced binary contradiction detection tasks that align with our theoretical framing. Further details are provided in Appendix A, where Table 3 presents example instances from each dataset.

SNLI The Stanford Natural Language Inference (SNLI) corpus (Bowman et al., 2015) is one of the most widely used human-annotated NLI benchmarks, and therefore serves as an important and challenging test case for assessing our correlation hypothesis. We exclude the training split to avoid potential overlap with the pretraining corpora of LLMs. From the validation and test sets, we construct two balanced subsets of 1,000 instances each through uniform sampling. The labels are mapped to a binary scheme: contradiction is assigned to the `True` class (label 1), while entailment and neutral are assigned to the `False` class (label 0).

bAbI We adapt Task 9 (Negation) from the bAbI dataset (Weston et al., 2015), a widely used benchmark designed to evaluate the reasoning and linguistic capabilities of language models. This task aligns with the definition of contradiction, where the premise entails the negation of the hypothesis. We transform the original question–answering format into premise-hypothesis pairs. Specifically, we construct four types of instances: Contradiction, Forward Entailment, Reverse Entailment, and Compatibility, following the natural logic taxonomy of MacCartney (2009). For instance, given the premise "*Mary is in the kitchen*", the hypothesis "*Mary is in the bedroom*" forms a Contradiction instance. Based on five locations and four actors, it contains 120 instances in total, including 60 `True` cases and 20 instances for each of the other three `False` types of instance, thus maintaining a balanced binary distribution.

Wikidata We construct three contradiction detection datasets from Wikidata (Vrandečić and Krötzsch, 2014). We include three different is-a relations, resulting in separate datasets: Wiki Capital (Wiki-C), Wiki Language (Wiki-L), and Wiki Software (Wiki-S). Each fact is represented as a sentence using a fixed template. For example, in Wiki-C, a non-contradictory statement is "*The capital of Malta is Valletta*", while a contradictory statement is formed by replacing the correct object with an incorrect alternative of the same type, such as "*The capital of Malta is Brasília*", as illustrated in Figure 1. Each dataset contains 100 pairs derived from 50 entities. Since the statements are grounded in factual knowledge, neutrality does not arise. The datasets are therefore naturally balanced between contradictions and non-contradictions.

5.2 Setup

Models We use models from the Llama (Touvron et al., 2023), Gemma (Team et al., 2024), and Qwen (Team, 2025) families. Their open-source nature (Wolf et al., 2020) allows direct access to the next-token probabilities, enabling explicit computation of surprisal via our token-to-word decoding algorithm. We use greedy decoding with deterministic settings (e.g., temperature) to ensure consistent probability estimation across models. The evaluated checkpoints include Llama-3.2-3B (`Llama`), Llama-3.2-3B-Instruct (`Llama-I`), Gemma-3-4B

(Gemma), and Qwen3-4B (Qwen). This configuration allows comparison of model behavior across architectures and parameter scales. See Appendix B for more details.

Evaluation We assess the relationship between surprisal and contradiction using two complementary metrics. First, we report threshold-based accuracy to provide an interpretable task-level perspective. Accuracy is computed on balanced datasets, and its statistical significance is verified by a one-sided binomial test. Second, following Merrill et al. (2024), we report ROC-AUC, to capture the overall strength of correlation without requiring a fixed decision threshold. Together, they offer both threshold-free and threshold-dependent views of the surprisal–contradiction relationship. To control for potential spurious correlations unrelated to semantics, we also include a Length baseline, which measures correlation using only hypothesis length (Gururangan et al., 2018).

Validation For threshold-based evaluation, the decision boundary on surprisal is determined from the SNLI validation set and applied across all datasets.[3] This setup keeps the evaluation consistent and comparable across domains without any dataset-specific tuning. It allows us to test the generality of the relationship between surprisal and contradiction, independent of dataset domain. The reported results should be regarded as a conservative estimate, as performance could be easily improved with dataset-specific hyperparameter tuning or model fine-tuning, which is not the goal of this work.

Settings We use TEMP context, Direct surprisal, and Mean aggregation in the main experiments, as this combination exhibits the most stable behavior across datasets and domains. This choice is also consistent with psycholinguistic findings that aggregated surprisal reflects distributed lexical uncertainty (Smith and Levy, 2013). We conduct a detailed analysis of each context construction, surprisal computation, and aggregation variant in Section 6.

5.3 Results

We now report the empirical results for both contradiction detection and correlation analysis.

[3] 4.0 for all models, except 5.0 for Qwen.

Model	Val.	SNLI	bAbI	Wiki-C	Wiki-L	Wiki-S
Length	52.5	50.1	45.0	50.0	50.0	50.0
Llama	63.9	62.4	59.2	97.0	91.0	97.0
Llama-I	63.7	61.5	61.7	93.0	90.0	99.0
Gemma	66.2	63.6	54.2	97.0	85.0	83.0
Qwen	66.5	64.3	72.5	99.0	99.0	100.0

Table 1: Accuracy (%) on contradiction detection under the TEMP context, Direct surprisal, and Mean aggregation. Thresholds are tuned on the SNLI validation set (Val) for each model and applied across datasets.

Model	Val.	SNLI	bAbI	Wiki-C	Wiki-L	Wiki-S
Length	48.3	47.5	52.7	53.3	49.4	51.7
Llama	69.0	66.7	62.5	100.0	100.0	100.0
Llama-I	68.5	66.8	64.7	99.6	100.0	100.0
Gemma	71.1	69.1	56.1	100.0	100.0	100.0
Qwen	71.1	69.6	86.9	98.1	100.0	100.0

Table 2: ROC-AUC (%) measuring the correlation between surprisal and contradiction under the TEMP context, Direct surprisal, and Mean aggregation. The Val column corresponds to the SNLI validation set.

Contradiction Detection Accuracies are summarized in Table 1. Across all datasets and models, accuracies are consistently above both the random baseline of 50% and the Length baseline, confirming that the surprisal-contradiction correlation generalizes to a practical decision setting. Performance also increases with model improvements over time, with Qwen consistently performing best. Notably, the threshold tuned on the SNLI validation set transfers effectively to the others without further adjustment, suggesting strong cross-domain consistency. The factual Wikidata sets approach ceiling performance, while the bAbI and SNLI data remain more challenging. All results are statistically significant under a one-sided binomial test compared to random chance ($p < 0.05$), except for Gemma on the bAbI dataset. This, in fact, reflects a relatively weaker cross-domain generalization from SNLI to bAbI in this specific case. Overall, these findings show that the thresholded surprisal signal can reliably distinguish between contradictory and non-contradictory sentence pairs.

Correlation Correlation results under the TEMP context, Mean aggregation, and Direct surprisal are reported in Table 2. With the only exception of Gemma on bAbI, all ROC-AUC scores surpass 60% across datasets and models, showing a clear positive correlation between surprisal and contradiction, especially on SNLI. These results comple-

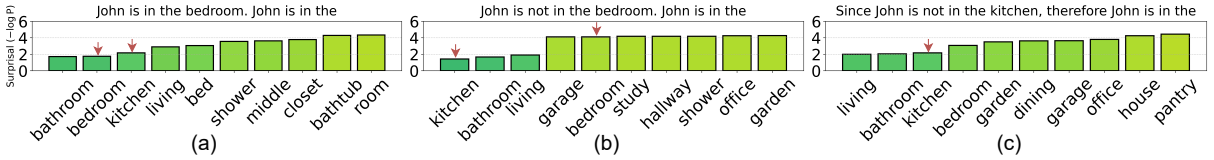

Figure 3: Surprisal distributions of Llama on bAbI examples. Red arrows highlight words whose surprisal is unexpectedly low or high, showing model bias and failed negation understanding.

ment the previous contradiction detection evaluation, and provide direct evidence for the positive correlation. Among models, Qwen again performs best. Across datasets, the correlation is strongest in the factual Wikidata sets and weaker, though still consistent, in bAbI and the human-annotated SNLI data. The Length baseline remains near random across datasets, confirming that the observed correlations are not driven by superficial artifacts.

6 Analysis and Discussion

The relatively low performance on the bAbI dataset motivates a closer error analysis.[4] Figure 3 presents representative examples illustrating two key issues: model bias driven by word correlation, and failure to handle negation.

Correlation over Causation A striking example of model bias towards correlation is shown in Figure 3 (a). Given "*John is in the bedroom. John is in the*", Llama assigns "*kitchen*" the third-lowest surprisal, even though the word creates a direct contradiction with the premise. However, when directly prompted with the same premise and hypothesis, the model correctly spots a contradiction.[5] This discrepancy indicates that the surprisal magnitude reflects not only contextual reasoning but also word correlation. That is, the model tends to favor words that are frequent in its training data and those present in the preceding context (Niu et al., 2025). Together, these tendencies suggest that LLM word probabilities may reflect correlation over causation.

Negation Understanding In the negation examples shown in Figure 3 (b) and (c), the model again behaves unexpectedly. Given the negated premise "*John is not in the kitchen.*" and the hypothesis "*John is in the kitchen*", it should assign "*kitchen*" in the hypothesis a high surprisal to reflect the obvious contradiction. However, even with TEMP context, Llama assigns the word the third-lowest surprisal. This suggests that the model does not properly account for negation when making predictions (Rezaei and Blanco, 2024), which is unexpected considering the simplicity of this example and the capabilities of modern LLMs.

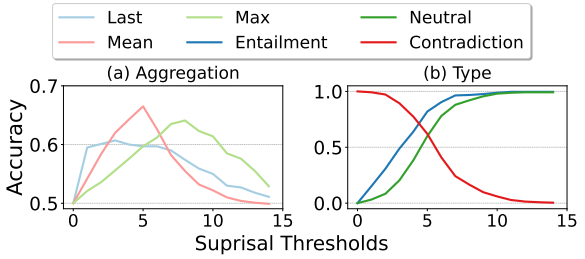

Figure 4: Threshold tuning on SNLI (Qwen) (a) by surprisal aggregation method, and (b) by type of instance using Mean aggregation.

Threshold Tuning Figure 4 shows how surprisal thresholds affect NLI accuracy. In (a), the aggregation methods produce distinct curves: Mean peaks near a threshold of 5, Max shifts the peak toward higher thresholds, and Last remains flatter overall. Despite these differences, all methods are able to exceed 60% accuracy across a broad range, demonstrating the robustness of the surprisal–contradiction correlation. Panel (b) reveals clear monotonic trends: the accuracy of entailment and neutral cases rise together with increasing threshold, showing nearly parallel curves, while contradiction accuracy declines symmetrically. The intersection of these trends aligns with the Mean peak in (a), confirming that the tuned threshold defines a stable surprisal-based decision boundary.

Counterfactual Context We examine how context construction affects surprisal-based contradiction detection, and how models respond to counterfactual premises. The factual Wiki-C dataset and its counterfactual variant Wiki-C* are used. In Wiki-C*, each pair is created by swapping the premise and hypothesis, such that in contradiction cases the new premise becomes objectively false.

[4] We observed a similar trend for both Direct and Relative, as shown in Table 4 in the Appendix.

[5] See Table 7 in the Appendix for the prompt we used.

For instance, the false statement "*The capital of Malta is Brasília*" serves as the premise, while the true statement "*The capital of Malta is Valletta*" becomes the hypothesis (Figure 1c). The `Mean` aggregation and `Direct` surprisal are controlled for comparison using the `Qwen` model. In Figure 5, both `CAT` and `TEMP` outperform `H-only` in the factual setting (a). `TEMP` achieves near-perfect accuracy at lower thresholds and across a wide range, showing that causal framing helps the model use premise information more effectively and strengthens the surprisal–contradiction link. `H-only` also reaches high accuracy, though at larger thresholds, suggesting that the model already encodes factual knowledge internally. In the counterfactual setting (b), accuracy drops for all cases. This indicates that internal factual knowledge interferes with reasoning when the premise is false. The decline is most evident for `H-only` and `CAT`, while `TEMP` maintains stronger performance, showing that structured prompting reduces reliance on internal knowledge and allows the model to attend more effectively to the given premise, even when it is objectively incorrect.

Distribution To examine whether the relationship remains stable under a naturally imbalanced label distribution, we conduct an additional analysis on a randomly resampled SNLI test set of 1,000 instances. Contradiction accounts for about 31% of the data, resulting in a majority baseline accuracy of approximately 69%. The experimental setup remains the same, using `Qwen` with `TEMP` context, `Direct` surprisal, and `Mean` aggregation. The surprisal threshold is re-tuned on a newly resampled SNLI validation set reflecting the same imbalanced distribution. Under this setup, accuracy reaches approximately 71%, still exceeding the majority baseline. More importantly, the ROC-AUC score increases to approximately 72%. Compared to the balanced test set, the correlation analysis remains stable and even improves. These results are consistent with our main findings on balanced datasets and further support the robustness of the observed surprisal-contradiction correlation under natural label distributions.

7 Conclusion

We examined whether lexical surprisal correlates with sentential contradiction. To support this analysis, we introduced a token-to-word decoding algorithm that extends theoretically grounded word probability estimation to open-vocabulary settings, enabling practical and accurate computation of surprisal in token-based language models. From the empirical results, we found a clear positive correlation between surprisal values and contradiction labels across multiple datasets and language models. We also demonstrated that methods based on this correlation hypothesis achieved consistent task performance significantly above random chance. The use of a single surprisal threshold further validated robustness and generalization of our findings across datasets and domains. To the best of our knowledge, our results provide the first systematic evidence that high surprisal is consistently associated with semantic inconsistency and contradiction. Our analysis also provides new insights into the limitations of LLMs: their tendency to prioritize correlation over causation, and their persistent struggles with negation understanding. We hope this research will inspire further work toward more transparent, explainable, and reliable LLMs.

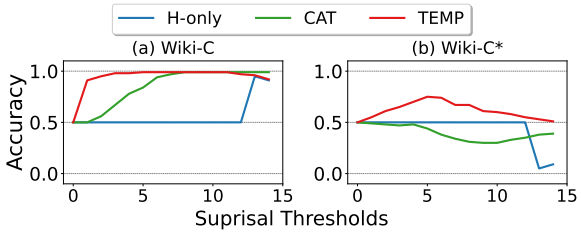

Figure 5: Accuracy across surprisal thresholds for three context constructions on Wiki-C and its counterfactual version Wiki-C*. Results are computed with `Qwen` using `Mean` aggregation and `Direct` surprisal.

Limitations

While our results establish a consistent correlation between surprisal and contradiction, several limitations remain. First, correlation does not imply causation: surprisal reflects statistical expectations rather than causal or logical reasoning, and thus can not fully explain how LLMs capture semantic inconsistency. Second, computing surprisal requires token-to-word decoding and open-vocabulary probability estimation, which can be time-consuming for large models or corpora and may limit scalability of our findings in practical applications. Third, the use of a surprisal threshold, although intended for illustrative purposes, introduces a tunable hyperparameter that may affect generalizability. Fourth, our current analysis aggregates multiple surprisal values into a single measure for correlation analysis.

Identifying the specific words that contribute most to contradiction would offer a more fine-grained and interpretable view of how semantic inconsistency arises. Finally, current models still exhibit weaknesses, such as bias toward surface correlation and difficulty with negation understanding. Future extensions of this study may depend on more advanced LLMs that can better address these limitations.

Acknowledgments

This research was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), and the Alberta Machine Intelligence Institute (Amii).

References

Steven Bird. 2006. NLTK: The Natural Language Toolkit. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 69–72, Sydney, Australia. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Cleo Condoravdi, Dick Crouch, Valeria de Paiva, Reinhard Stolle, and Daniel G. Bobrow. 2003. Entailment, intensionality and text understanding. In *Proceedings of the HLT-NAACL 2003 Workshop on Text Meaning*, pages 38–45.

Ido Dagan, Bill Dolan, Bernardo Magnini, and Dan Roth. 2010. Recognizing textual entailment: Rational, evaluation and approaches–erratum. *Natural Language Engineering*, 16(1):105–105.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Machine learning challenges workshop*.

Marie-Catherine de Marneffe, Anna N. Rafferty, and Christopher D. Manning. 2008. Finding contradictions in text. In *Proceedings of ACL-08: HLT*, pages 1039–1047, Columbus, Ohio. 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.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. 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*.

Sanda Harabagiu, Andrew Hickl, and Finley Lacatusu. 2006. Negation, contrast and contradiction in text processing. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1*, AAAI'06, page 755–762. AAAI Press.

Bradley Hauer and Grzegorz Kondrak. 2022. WiC = TSV = WSD: On the equivalence of three semantic tasks. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2478–2486, Seattle, United States.

Bradley Hauer and Grzegorz Kondrak. 2023. Taxonomy of problems in lexical semantics. In *Findings of the Association for Computational Linguistics: ACL 2023*.

Jie Huang and Kevin Chen-Chuan Chang. 2023. Towards reasoning in large language models: A survey. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1049–1065, Toronto, Canada. Association for Computational Linguistics.

Jerrold J. Katz. 1972. *Semantic Theory*. Harper & Row, New York.

Andrey Nikolaevich Kolmogorov. 1933. Foundations of probability theory. *Julius Springer: Berlin, Germany*.

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

Bill MacCartney. 2009. *Natural language inference*. Stanford University.

Bill MacCartney and Christopher D. Manning. 2009. An extended model of natural logic. In *Proceedings of the Eight International Conference on Computational Semantics*, pages 140–156, Tilburg, The Netherlands. Association for Computational Linguistics.

Siddarth Mamidanna, Daking Rai, Ziyu Yao, and Yilun Zhou. 2025. All for one: Llms solve mental math at the last token with information transferred from other tokens. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

William Merrill, Alex Warstadt, and Tal Linzen. 2022. Entailment semantics can be extracted from an ideal language model. In *Proceedings of the 26th Conference on Computational Natural Language Learning (CoNLL)*, pages 176–193, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

William Merrill, Zhaofeng Wu, Norihito Naka, Yoon Kim, and Tal Linzen. 2024. Can you learn semantics through next-word prediction? the case of entailment. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2752–2773, Bangkok, Thailand. Association for Computational Linguistics.

George A Miller. 1995. Wordnet: a lexical database for english. *Comm. ACM*, 38(11).

Yixin Nie, Haonan Chen, and Mohit Bansal. 2019. Combining fact extraction and verification with neural semantic matching networks. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Made Nindyatama Nityasya, Haryo Wibowo, Alham Fikri Aji, Genta Winata, Radityo Eko Prasojo, Phil Blunsom, and Adhiguna Kuncoro. 2023. On "scientific debt" in NLP: A case for more rigour in language model pre-training research. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8554–8572, Toronto, Canada. Association for Computational Linguistics.

Jingcheng Niu, Xingdi Yuan, Tong Wang, Hamidreza Saghir, and Amir H. Abdi. 2025. Llama see, llama do: A mechanistic perspective on contextual entrainment and distraction in LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16218–16239, Vienna, Austria. Association for Computational Linguistics.

Byung-Doh Oh and William Schuler. 2024. Leading whitespaces of language models' subword vocabulary pose a confound for calculating word probabilities. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3464–3472, Miami, Florida, USA. Association for Computational Linguistics.

Tiago Pimentel and Clara Meister. 2024. How to compute the probability of a word. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18358–18375, Miami, Florida, USA. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, and 1 others. 2018. Improving language understanding by generative pre-training. Technical report, OpenAI.

MohammadHossein Rezaei and Eduardo Blanco. 2024. Paraphrasing in affirmative terms improves negation understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 602–615, Bangkok, Thailand. Association for Computational Linguistics.

Alan Ritter, Stephen Soderland, Doug Downey, and Oren Etzioni. 2008. It's a contradiction – no, it's not: A case study using functional relations. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 11–20, Honolulu, Hawaii. Association for Computational Linguistics.

Claude E Shannon. 1948. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423.

Ning Shi, Bradley Hauer, and Grzegorz Kondrak. 2024a. Lexical substitution as causal language modeling. In *Proceedings of the 13th Joint Conference on Lexical and Computational Semantics (*SEM 2024)*, pages 120–132, Mexico City, Mexico. Association for Computational Linguistics.

Ning Shi, Bradley Hauer, Jai Riley, and Grzegorz Kondrak. 2024b. Paraphrase identification via textual inference. In *Proceedings of the 13th Joint Conference on Lexical and Computational Semantics (*SEM 2024)*, pages 133–141, Mexico City, Mexico. Association for Computational Linguistics.

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

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. 2023. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051, Toronto, Canada. Association for Computational Linguistics.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, and 89 others. 2024. Gemma: Open models based on gemini research and technology. *Preprint*, arXiv:2403.08295.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Simone Tedeschi, Johan Bos, Thierry Declerck, Jan Hajič, Daniel Hershcovich, Eduard Hovy, Alexander Koller, Simon Krek, Steven Schockaert, Rico Sennrich, Ekaterina Shutova, and Roberto Navigli. 2023. What's the meaning of superhuman performance in today's NLU? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12471–12491, Toronto, Canada. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Johan van Benthem. 2008. A brief history of natural logic. In Mihir K. Chakraborty, Mohua Nath Mitra, Benedikt Löwe, and Sundar Sarukkai, editors, *Logic, Navya-Nyāya & Applications: Homage to Bimal Krishna Matilal*, pages 21–42. College Publications, London. Presented at the International Conference on Logic, Navya-Nyāya & Applications (ICLNNA), Kolkata, India.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M. Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. 2015. Towards ai-complete question answering: A set of prerequisite toy tasks.

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.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi 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. Transformers: State-of-the-art natural language processing. In *EMNLP: System Demonstrations*.

Zijun Wu, Zi Xuan Zhang, Atharva Naik, Zhijian Mei, Mauajama Firdaus, and Lili Mou. 2023. Weakly supervised explainable phrasal reasoning with neural fuzzy logic. In *The Eleventh International Conference on Learning Representations*.

A Data

Table 3 presents example premise–hypothesis pairs from each dataset used in our experiments, including SNLI, Wikidata, and bAbI. We illustrate one instance of each type, together with its corresponding binary contradiction label in our setting. All datasets are used solely for non-commercial, academic research and contain no personal or identifiable information. The datasets are entirely in English and span diverse domains, including image captions (SNLI), factual relations (Wikidata), and synthetic reasoning narratives (bAbI). We verified that they contain no personally identifying or offensive content through random sample inspection and by referencing the published documentation and licensing standards. Datasets constructed or processed in this work have been released publicly on GitHub.[6]

SNLI The Stanford Natural Language Inference (SNLI) corpus (Bowman et al., 2015) is the most widely used human-annotated NLI benchmark, making it both an important and a challenging test case for assessing whether surprisal–contradiction correlation holds on real data. It contains about 570,000 premise–hypothesis pairs based on image captions, each labeled as entailment, contradiction, or neutral. The dataset is divided into a training set of roughly 550,000 pairs and two evaluation sets of 10,000 pairs each for validation and test. A publicly available version is hosted on HuggingFace Datasets[7], under the Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0) license. For our study, we adapt SNLI to a binary contradiction detection task to examine the correlation between surprisal values and contradiction labels. We exclude the training split because it may overlap with the pretraining corpora of LLMs and could bias our evaluation. Instead, we use only the validation and test splits, which together contain 20,000 examples. The original three-way labels are mapped to a binary scheme. Contradiction is assigned to the True class (label 1), while both entailment and neutral are assigned to the False class (label 0). From each of the validation and test sets, we uniformly sample 1,000 instances, balanced between 500 contradictions and 500 non-contradictions. This construction converts SNLI, the standard benchmark for NLI, into a binary contradiction detection dataset that enables us to test our methods on a large-scale, human-annotated corpus.

bAbI We adopt the bAbI dataset (Weston et al., 2015), a benchmark designed to evaluate the reasoning and linguistic capabilities of language models, released by Facebook AI Research under the BSD 3-Clause License. Among the 20 tasks that comprise the dataset, we focus on its Task 9, Negation, as it is particularly relevant to contradiction detection. This task directly follows our definition of contradiction, where the premise entails the negation of the hypothesis. The bAbI authors do not provide a fixed dataset of instances, but rather a program to generate as many instances as needed. Each generated instance consists of a fact, which is a statement describing an action or location, a yes/no question, and the answer. To construct our dataset, we transform these into contradiction detection instances, i.e., premise-hypothesis pairs. We construct four types of relations following the natural logic taxonomy proposed by MacCartney (2009). *Contradiction* instances are derived from the original bAbI Task 9 labels, where an answer of "*No.*" indicates that the fact does not support the question. We convert these into premise–hypothesis form by taking the fact as the premise and rewriting the question as a declarative sentence for the hypothesis. This yields pairs where the hypothesis contradicts the premise. *Forward entailment* instances are created by generalizing the hypothesis, replacing its location with the category label "*in the house*" for indoor locations or "*not in the house*" for outdoor locations. *Reverse entailment* instances are generated by instead generalizing the premise in the same way and leaving the hypothesis as the original fact. *Compatibility* instances are formed by replacing the actor in the hypothesis with a different individual from the premise, ensuring that the two statements are consistent but independent. The dataset is generated from a controlled domain of 5 locations and 4 actors. We create 60 contradiction instances and 20 instances of each of the other three relation types, for a total of 120 examples. This distribution yields a balanced binary contradiction detection dataset, while still covering the full range of natural logic relations. Each instance is then relabeled with a binary contradiction label, where only contradiction is marked as `True` (label 1). The entire construction process is fully automated, starting from the

[6] github.com/ShiningLab/CON2LM
[7] huggingface.co/datasets/stanfordnlp/snli

Data	Premise (P)	Hypothesis (H)	Type	CON Label
bAbI	Daniel is in the bathroom. Mary is in the playground. Mary is not in the house. Daniel is in the garden.	Daniel is in the bedroom. Mary is not in the house. Mary is in the driveway. Mary is in the garden.	Contradiction Forward Entailment Reverse Entailment Compatibility	1 (`True`) 0 (`False`) 0 (`False`) 0 (`False`)
Wiki Capital	The capital of Malta is Valletta. The capital of Malta is Valletta.	The capital of Malta is Brasília. The capital of Malta is Valletta.	Contradiction Entailment	1 (`True`) 0 (`False`)
Wiki Language	The official language of the country Norway is Norwegian. The official language of the country Norway is Norwegian.	The official language of the country Norway is Russian. The official language of the country Norway is Norwegian.	Contradiction Entailment	1 (`True`) 0 (`False`)
Wiki Software	The company that developed Exchange ActiveSync is called Microsoft. The company that developed Exchange ActiveSync is called Microsoft.	The company that developed Exchange ActiveSync is called Macromedia. The company that developed Exchange ActiveSync is called Microsoft.	Contradiction Entailment	1 (`True`) 0 (`False`)
SNLI	A child climbing a rock. This is a man in a photo booth. A motorcycle races.	An old woman sits on a rock. A man in a photo booth. A bike is racing a cheetah.	Contradiction Entailment Neutral	1 (`True`) 0 (`False`) 0 (`False`)

Table 3: Example instances from the datasets used in our experiments. Each row shows a premise–hypothesis pair, the associated type of instance, and the corresponding binary contradiction detection (CON) label.

generation of bAbI instances via code provided by the authors of bAbI[8].

Wikidata We construct three contradiction detection datasets using Wikidata (Vrandečić and Krötzsch, 2014), a collaboratively curated knowledge graph of real-world entities, under the Creative Commons CC0 License. Each fact is verbalized into English with a fixed natural language template, producing a non-contradiction instance. To generate contradictions, we replace the correct object with an incorrect alternative of the same type, sampled at random under a fixed seed. We focus on three distinct relation types, resulting in three separate datasets: Wiki Capital (Wiki-C), Wiki Language (Wiki-L), and Wiki Software (Wiki-S). All sentences are strictly templated in English. The corresponding templates are:

- "*The capital of {country} is {city}.*"
- "*The official language of the country {country} is {language}.*"
- "*The company that developed {software} is called {company}.*"

Entities are selected so that both the correct object and its replacement are single words, allowing surprisal to be computed cleanly at the word level. For each relation type, we generate 100 sentence pairs spanning 50 entities, with each entity contributing both a true and a false hypothesis. Since the statements are grounded in factual world knowledge, neutrality does not arise: each premise–hypothesis pair is labeled as True (label 1) or False (label 0).

B Models

We use models from the Llama (Touvron et al., 2023), Qwen (Team, 2025), and Gemma (Team et al., 2024) families, all trained for next-token prediction under a causal language modeling objective. Their open-source nature (Wolf et al., 2020) allows direct access to token-level probabilities, enabling explicit computation of the surprisal values via our token-to-word decoding. We retrieve probabilities via greedy decoding to ensure consistent probability comparisons. The decoding configuration (e.g., temperature) is kept fully deterministic by disabling stochastic sampling and using greedy decoding for probability retrieval. The following checkpoints are used: Llama-3.2-3B[9], Llama-3.2-3B-Instruct[10], Gemma-3-4B[11], and Qwen3-4B[12]. Each model is evaluated using its native tokenizer to ensure consistency between subword segmentation and probability estimation. All experiments are run on a single NVIDIA GeForce RTX 3090

[8] github.com/facebookarchive/bAbI-tasks

[9] huggingface.co/meta-llama/Llama-3.2-3B

[10] huggingface.co/meta-llama/Llama-3.2-3B-Instruct

[11] huggingface.co/google/gemma-3-4b-pt

[12] huggingface.co/Qwen/Qwen3-4B

Model	SNLI			bAbI			Wiki-C			Wiki-L			Wiki-S		
	Last	Mean	Max	Last	Mean	Max	Last	Mean	Max	Last	Mean	Max	Last	Mean	Max
Llama	61/60	67/66	65/65	82/83	62/63	56/58	98/98	100/99	100/99	100/100	100/100	92/99	100/100	100/100	100/100
Llama-I	61/60	67/66	64/64	85/85	65/66	63/63	98/98	100/99	99/99	100/100	100/100	96/100	100/100	100/100	100/100
Gemma	62/61	69/69	68/68	79/75	56/53	58/54	100/100	100/100	100/100	100/100	100/100	100/100	100/100	100/100	100/100
Qwen	62/61	70/69	68/67	92/92	87/87	84/84	98/98	98/99	98/99	100/100	100/100	100/100	100/100	100/100	100/100
Length		47.47			53.32			49.44			51.64			52.74	

Table 4: ROC-AUC (%) measuring the correlation between surprisal and contradiction under the TEMP context. Each cell reports Direct / Relative surprisal results for three aggregation methods (Last, Mean, Max).

PoS Category	Tags Included
Nouns	NN, NNS, NNP, NNPS
Verbs	VB, VBD, VBG, VBN, VBP, VBZ
Adjectives	JJ, JJR, JJS
Adverbs	RB, RBR, RBS

Table 5: Part-of-speech tags of content words.

Models	QA	SC
Llama	98	92
Llama-I	98	92
Gemma	88	90
Qwen	90	90

Table 6: Accuracy (in %) on factual knowledge probing using Question Answering (QA) and Sentence Completion (SC) prompts from Wiki-Capital (Wiki-C).

GPU using a unified loading process through the Hugging Face Transformers v4.51.3 pipeline. This configuration provides a consistent environment for surprisal estimation across different model architectures and parameter scales. Regarding licenses, Llama is provided under the Llama 3.2 Community License Agreement; Gemma under the Gemma Terms of Use; and Qwen under the Apache License 2.0. They are employed solely for evaluation within research prototypes, consistent with their community and open-source licenses. No models or data derivatives are deployed beyond research contexts.

C Setup

Content Word We compute surprisal only over content words. To identify these words, we use part-of-speech (POS) tags assigned by the NLTK v3.9.1 tagger (Bird, 2006). Words belonging to the categories listed in Table 5 are retained, while all others (e.g., determiners, prepositions, pronouns, and auxiliary verbs) are excluded. In addition, common function words are filtered out using the English stopword list provided by NLTK.

Token-to-Word Decoding For open-vocabulary word probability estimation, our token-to-word decoding algorithm uses constrained beam search parameterized by the beam width W and beam depth D. We set $W = 10$ and $D = 10$, following empirical guidance from the Princeton WordNet (Miller, 1995). While larger values generally improve coverage, they also increase computation time. To ensure efficiency across large-scale experiments, we adopt these moderate settings as a practical trade-off. When the target word is known (closed-vocabulary computation), beam search is unnecessary and probability estimation reduces to a single forward pass. As illustrated in Figure 2, this decoding procedure enables accurate word-level surprisal estimation and supports open-vocabulary retrieval from the next-word distribution.

D Knowledge Availability

Before evaluating contradiction detection, we first verify that the models possess the relevant factual knowledge. This ensures that performance differences observed in the counterfactual analysis in Section 6 reflect reasoning behavior rather than lack of information. We test two prompting formats: Question Answering (QA) and Sentence Completion (SC), using the templates defined in Table 7. Accuracy is measured by checking whether the correct fact appears in the model's response. Table 6 show that all models achieve high accuracy across both QA and SC prompts on the Wiki-Capital (Wiki-C) dataset, confirming that they have sufficient access to factual knowledge. QA generally yields slightly higher accuracy than SC, and models in the Llama family perform best overall. These results confirm that models already encode the necessary world knowledge, supporting our interpretation that performance degradation in Wiki-C* arises from counterfactual reasoning interference rather than ignorance of factual information.

CON Prompt
Given two sentences, your job is to determine if the second sentence contradicts the first sentence. Respond with True if there is a contradiction and False if there is not. Only return True or False without explanation. Sentence 1: "{P}" Sentence 2: "{H}" Your answer:
CAT Prompt
Given a sentence, your job is to determine if it contains a contradiction. Respond with True if there is a contradiction and False if there is not. Only return True or False without explanation. Sentence: "{P $\circ$ H}" Your answer:
TEMP Prompt
Given a sentence, your job is to determine if it contains a contradiction. Respond with True if there is a contradiction and False if there is not. Only return True or False without explanation. Sentence: "Since {P}, therefore {H}" Your answer:
Question Answering (QA) Prompt
Question: What is the capital of {country}? Answer:
Sentence Completion (SC) Prompt
The capital of {country} is

Table 7: Prompts used to query LLMs. The first three are used for contradiction detection. The last two are used for factual knowledge probing in the Wiki-Capital (Wiki-C) dataset.