PARSE: A Framework for Evaluating Linguistic and Typographical Perturbation Bias in Large Language Models

John Lacalamita

A Thesis Submitted to the Faculty of Graduate Studies
In Partial Fulfillment of the Requirements
for the Degree of Master of Science

Graduate Program in
Electrical Engineering and Computer Science
York University
Toronto, Ontario

April 2026

Abstract

Large language models (LLMs) are increasingly becoming ubiquitous in day-to-day tasks. Yet, despite the growing dependence on LLM-based systems, their sensitivity to surface-level linguistic variation has received little attention. We introduce PARSE (Prompt Alteration Response-Shift Evaluation), a modular framework that generates grammatical, typographical, and dialectal prompt variants, queries LLMs under identical conditions, and measures distributional output shifts. We apply PARSE to two case studies, film recommendation and privacy bias evaluation, across three models (GPT-4o-mini, Llama 3.2, DeepSeek-7B). Results show that output shifts scale with perturbation intensity: grammatical rewrites produce minimal effects, while typographical noise and dialect rewrites significantly alter recommendations and appropriateness ratings. Perturbations push film recommendations toward higher-rated, generic titles, and shift privacy ratings toward more restrictive values. These effects are directionally consistent across models, demonstrating that the linguistic form of a prompt systematically biases LLM outputs.

Acknowledgments

I extend my deepest gratitude to Dr. Yan Shvartzshnaider for his continued guidance throughout this project and in my growth as a researcher. I would also like to thank my family and friends for their unwavering support throughout this process. I am grateful to my classmates Ronny and Jazdeep for making this journey fun. Finally, to my girlfriend Mayu, thank you for always keeping me happy and laughing. This thesis would not be what it is today without all of you.

Contents

List of Tables

List of Figures

1 Introduction

LLM-powered platforms are rapidly displacing traditional search engines and curated databases as the primary way users seek information, health advice, and product recommendations. Despite the widespread use of chatbots such as ChatGPT and Gemini for a range of critical tasks, growing evidence shows that LLMs are susceptible to prompt sensitivity and can produce biased, inaccurate, or even harmful responses [1–3]. These issues can have grave consequences, as with several documented cases linking chatbot advice to a teenager's death [4, 5]. While prior work has examined how linguistic features of a user's input, such as dialect, grammar, and orthographic choices, affect NLP system performance, existing studies have focused primarily on classification accuracy in earlier architectures rather than on the distributional behavior of modern generative models.

Written prompts can implicitly signal aspects of a user's linguistic or social background through dialectal cues and grammatical patterns, and models such as BERT and T5 have been shown to degrade under dialectal and orthographic variation [6, 7]. African American Language text, for instance, has been shown to be disproportionately flagged as toxic by content moderation systems [8]. If models respond differently to semantically equivalent requests that differ only in linguistic form, then the quality of service a user receives may depend not on *what* they ask but on *how* they phrase it. Understanding whether and how such linguistic variation alters LLM outputs is therefore important for evaluating the robustness and consistency of these systems.

Furthermore, certain alignment techniques may amplify these disparities. In reinforcement learning from human feedback (RLHF), preference-trained reward models such as those studied by Mire et al. [9] penalize non-standard dialect outputs, effectively nudging aligned models toward Standard American English. More broadly, Ryan et al. [10] show that alignment disproportionately boosts performance for US English over other varieties, widening the gap between dialects from approximately 1% before alignment to as high as 17.1% afterward. Evaluation methodologies such as CheckList [11] and multi-prompt protocols [12] have further shown that semantically equivalent rephrasings can shift model scores and rankings.

Building on these lines of work, which have largely focused on classification accuracy or single-output correctness, we instead measure how surface-level linguistic variation shifts the *distribution* of generative outputs, such as recommendation lists or privacy biases [13], produced by modern LLMs. This thesis investigates the following research questions:

RQ1 What are the effects of different linguistic variants on the LLM outputs?

RQ2 What are the patterns in LLM outputs of different linguistic perturbations?

RQ3 What are the linguistic perturbation effects on different LLMs?

To address these questions, we introduce a modular **Prompt Alteration Response-Shift Evaluation (PARSE)** framework to measure distributional changes in generative outputs.

Our contributions are as follows. We present:

- A modular PARSE framework design for a task-agnostic pipeline that produces prompt variants using systematic grammatical rewrites (via Multi-VALUE [6]), classic typographical error taxonomies [14, 15], and LLM-generated dialect translations. PARSE systematically generates meaning-preserving grammatical, typographical, and dialectal variants of input prompts, queries multiple LLMs under identical conditions, and measures how outputs shift within each prompt pair.

- Evaluation of two case studies using the PARSE framework. The first case study focuses on film recommendation, examining how prompt variation alters the rating, genre composition, and maturity rating of recommended titles. The second study evaluates appropriateness levels of stated information flows in an education context using the Contextual Integrity (CI) framework [16]. In both case studies, we examine whether linguistic perturbations such as grammatical and typographical variants affect outputs (**RQ1**), whether different perturbations produce larger and directionally consistent shifts (**RQ2**), and whether these effects replicate across models (**RQ3**).

- Evaluation across three different LLMs (GPT-4o-mini, Llama 3.2, and DeepSeek-7B). Our results reveal that prompts containing common typos or written in a non-standard dialect return systematically different responses in both case study tasks.

The remainder of the thesis is organized as follows. Section 2 reviews related work on input variation, LLM-based recommendation evaluation, and contextual integrity. Section 3 describes the PARSE framework. Sections 5 and 6 present the film recommendation and privacy bias case studies, respectively. Section 7 discusses implications and limitations, and Section 8 concludes.

2 Background and Related Work

In this section, we review prior work on how language variation and prompt design affect the outputs of natural language processing models and recommender systems. A substantial body of research examines how differences in input form, including dialectal, grammatical, and orthographic variation, affect model robustness to perturbations. We organize the review around four areas. First, we provide a brief primer on large language model architecture and post-training, which grounds the mechanisms discussed in the rest of the review. Second, we survey work on input variation and prompt sensitivity, focusing on how dialects, grammar, and typographical errors can shift model behavior in classification and generation tasks. Third, we review work on LLM-based recommender systems, focusing on how these systems are evaluated for recommendation quality and related distributional concerns. Fourth, we introduce CI as a framework for privacy assessment and review how it has been applied to evaluate information-flow norms in both human and automated settings. Specific related work for LLM-based recommendation and CI is discussed in Sections 2.3 and 2.4, respectively.

2.1 LLM Primer

A large language model comprises a stack of transformer layers that together map an input string to a probability distribution over candidate next tokens. Each layer applies an attention mechanism that lets every position in the input attend to every other position, so the representation of any given token is shaped by its surrounding context rather than by the token alone.

As part of prompt processing, the model first tokenizes the input string into a sequence of subword tokens, each token is then mapped to a vector called an embedding. Attention layers then combine these embeddings so that each position's representation reflects both what the token is and where it sits relative to the rest of the prompt. At each generation step, the model computes a probability distribution over candidate next tokens conditioned on this representation, and a continuation is selected from that distribution. If two prompts are semantically equivalent for a human reader but segmented differently into tokens, the model treats them as two different inputs from the first layer onward, and the resulting next-token distributions can differ even when the prompts carry identical meaning.

Fine-tuning and alignment techniques. Pretraining produces a model that completes text, but it does not produce a model that follows assistant-style instructions. The main fine-tuning approaches include supervised fine-tuning (SFT), which trains the model on demonstrations of desired assistant behavior, reinforcement learning from human feedback (RLHF) [17], which trains a separate reward model on human preference judgments and then updates the base model to maximize reward, typically through PPO, and direct preference optimization (DPO) [18], which achieves a similar effect without training a separate reward model by directly adjusting the policy's likelihood of preferred versus dispreferred outputs on the same preference pairs. Rejection sampling draws many candidate completions during fine-tuning and retains only those that pass a filtering criterion. These techniques are relevant to PARSE because preference-based alignment can itself encode biases toward some linguistic forms over others.

Prior work shows that preference models can assign lower rewards to non-standard dialect outputs and that alignment can widen performance gaps across dialects [9, 10]. A model whose post-training has more aggressively rewarded standardized assistant outputs may therefore drift more strongly toward those outputs when faced with non-standard input, independently of any effect from tokenization.

2.2 Input Variation and Prompt Sensitivity in Natural Language Processing Systems

Before the emergence of modern large language models, a substantial body of work studied the sensitivity of earlier natural language processing (NLP) architectures to input variation. These studies are relevant because they establish that model performance degrades under non-standard inputs, a vulnerability that motivates our investigation of whether the same patterns persist in the generative LLM setting.

2.2.1 Dialectal and Grammatical Robustness

BERT [19], a masked language model that learns bidirectional representations of text through pre-training on large corpora, and T5 [20], a sequence-to-sequence model that frames all NLP tasks as text-to-text problems, both exhibit sizable accuracy drops on non-standard English inputs compared to Standard American English (SAE). Ziems et al. [6] demonstrated this vulnerability using their Multi-VALUE framework, which evaluates downstream tasks including question answering, machine translation, and semantic parsing. Multi-VALUE generates dialectal variants by applying rule-based grammatical transformations to SAE text. Specifically, the framework implements perturbation rules for 189 grammatical features drawn from the Electronic World Atlas of Varieties of English (eWAVE) [21]. eWAVE is a typological database that records 235 grammatical features observed across 77 English varieties. Multi-VALUE operationalizes 189 of these 235 features as executable rewrite rules. Our PARSE framework relies on the same set of 189 grammatical features as its rewrite inventory (see Section 3).

Prior work has also shown that models can exhibit downstream disparities, such as disproportionately flagging African American Language (AAL) text as toxic [8]. Furthermore, Deas et al. [22] investigated whether models can accurately comprehend and generate dialectal text. Focusing on intrinsic generative capabilities, they evaluate models on tasks like counterpart generation (translating text between SAE and AAL) and masked span prediction (filling in missing words within a dialectal sentence). They find that models struggle to accurately produce AAL text compared to SAE, revealing that dialectal disparities are embedded in the learned representations of modern architectures.

Beyond training data and model architecture, dialectal bias can also be introduced through the alignment process, in which a language model's behavior is tuned to better reflect human preferences. In reinforcement learning from human feedback (RLHF), a reward model scores candidate outputs to guide the LLM toward preferred behaviors. Mire et al. [9] find that preference models assign lower rewards to AAL-style outputs than to equivalent SAE outputs, effectively nudging an aligned model toward the standard dialect. Ryan et al. [10] evaluate language models on a dialogue intent prediction task involving American, Nigerian, and Indian English speakers. They find that while alignment procedures improve overall task performance, they disproportionately improve performance for US English, increasing the disparity between English dialects from approximately 1% before alignment to as high as 17.1% afterward. These works suggest that dialectal disparities can arise not only from training data, but also from alignment procedures that reward standard forms more strongly than non-standard ones.

2.2.2 Orthographic and Typographical Robustness

While the preceding section examined how grammatical and dialectal variation affects model behavior, a parallel line of research has studied how character-level noise, such as misspellings, transposed letters, and keyboard errors, disrupts NLP model performance. This work is relevant because typographical errors are among the most common forms of input variation in real-world user queries,

and understanding their effects on earlier models helps motivate our evaluation of whether modern LLMs exhibit similar vulnerabilities.

Kumar et al. [7] show that common spelling errors, including misspellings, character transpositions, and character insertions, disrupt subword tokenization in models such as BERT, reducing task accuracy on sentiment classification and semantic textual similarity benchmarks. Most modern language models do not process text as whole words; instead, they segment input into subword tokens using algorithms such as byte pair encoding (BPE) [23]. A single misspelling can cause the tokenizer to split a familiar word into multiple unfamiliar subword units, degrading the model's ability to recognize the intended word. Kumar et al. systematically introduce noise at varying intensities and demonstrate that even low error rates produce measurable accuracy drops due to this tokenization disruption.

Tasawong et al. [24] address the problem of typographical errors from the perspective of information retrieval. When users submit misspelled search queries, standard dense retrieval models often fail to match them with relevant documents. Tasawong et al. train dense retrieval models to handle typical typos, such as letter swaps and character omissions, by learning to map misspelled queries and their correct forms to nearby points in a shared embedding space. While this data augmentation approach improves retrieval accuracy on the types of spelling errors seen during training, the authors note that such methods face challenges in generalizing to out-of-distribution or unseen error types that were not represented in the training data.

An alternative approach to handling character-level noise is to avoid subword tokenization entirely. Subword segmentation is a preprocessing step in which text is split into units smaller than words (e.g., "playing" might be split into "play" and "##ing"), allowing models to handle a large vocabulary with a fixed token set. However, this segmentation is brittle to misspellings because a single changed character can alter the segmentation. Token-free byte-level models such as ByT5 [25] bypass this step by operating directly on raw bytes, making them inherently more tolerant to spelling variation, though at the cost of longer input sequences and higher computational requirements. Zheng and Saparov [26] take a different approach, augmenting few-shot prompts with perturbed examples during inference. They find that this data-augmented prompting improves robustness on many lexical changes such as synonym swaps, though typos can still degrade reasoning on more complex tasks.

2.2.3 Prompt Sensitivity and Evaluation Methodology

While the preceding sections focus on how models respond to dialectal and typographical variation in the input text itself, a related body of work examines the sensitivity of models to the way a task is presented, even if the underlying request is the same. This section reviews methods for systematically testing model behavior under controlled input variation.

Ribeiro et al. [11] introduce CheckList, a task-agnostic testing methodology inspired by software engineering practices. CheckList defines a set of behavioral tests, including minimum functionality tests, invariance tests, and directional expectation tests, that probe whether a model handles basic

linguistic phenomena correctly. Ribeiro et al. apply CheckList to commercial sentiment analysis, question answering, and duplicate-question detection systems, revealing predictable failure modes (such as failing on negation or named entity swaps) that are invisible under aggregate accuracy metrics. Kiela et al. [27] extend this idea with Dynabench, a platform for dynamic, human-in-the-loop benchmarking in which annotators iteratively create examples that fool a target model. Dynabench reveals that small rewording or negation changes can reliably break models even as overall benchmark scores continue to climb.

Mizrahi et al. [12] demonstrate that LLM evaluation results can vary substantially across semantically equivalent prompts. They evaluate multiple models on standard benchmarks using paraphrased versions of the same questions and find that the choice of prompt wording can change both absolute scores and relative model rankings. Their findings motivate multi-prompt evaluation protocols in which each test item is presented in several phrasings to obtain a more stable estimate of model performance.

Prior work across dialectal, orthographic, and prompt-sensitivity research typically defines robustness as the preservation of task accuracy, ensuring a model returns the same label despite minor input perturbations. While this definition suits classification tasks with a single expected output, it does not capture how surface-level linguistic variations might alter the broader distribution of outputs produced by generative models. Our framework addresses this gap. Rather than just evaluating whether a model returns the correct label, we test how wording changes shift the composition of generative outputs. By doing so, we extend robustness evaluation beyond traditional benchmarks, specifically measuring how consistently a model ranks recommendation lists or assigns appropriateness ratings when faced with semantically equivalent but linguistically diverse inputs.

2.3 LLM-Based Recommender Systems Evaluation

This section reviews work on how large language models have been evaluated as recommender systems, covering both recommendation quality and concerns about how recommendations are distributed across users or items.

Hua et al. [28] focus on user-side fairness, meaning that recommendations should not change unfairly across demographic groups such as gender or age. They introduce UP5, a framework for counterfactually fair prompting in which extra instructions or debiasing examples are added to the user's query. By training the model with fairness-aware prompts, they improve recommendation equity across groups with only a small drop in accuracy. Wang et al. [29] link the LLM to an external knowledge graph so that recommendations are grounded in factual item information, such as genre, release year, or popularity statistics, rather than relying solely on the model's parametric knowledge. By retrieving structured item data at inference time, the system can be more easily controlled and audited. Both methods adjust the system's behavior after generation or by adding extra context, rather than changing how the model itself forms recommendations.

Hou et al. [30] frame recommendation as a conditional ranking task and evaluate GPT-3.5 and GPT-4 as zero-shot rankers on MovieLens and Amazon datasets, finding competitive performance

but systematic position and popularity biases. Dai et al. [31] compare point-wise, pair-wise, and list-wise prompting strategies for ChatGPT across four recommendation domains and show that list-wise ranking achieves the best cost–performance trade-off. These studies characterize *what* LLMs recommend and *how well*, but do not test whether the *linguistic form* of the input prompt alters what is recommended.

While recent studies establish how modern LLMs perform as zero-shot rankers, evaluating the downstream impact of these generated lists on item providers requires metrics beyond standard accuracy. Biega et al. [32] formalize amortized fairness, arguing that over time all items should receive attention roughly proportional to their relevance. Singh and Joachims [33] build on this idea by establishing frameworks for exposure fairness in static rankings. Shifting the focus from producer exposure to long-term user well-being, Singh et al. [34] extend recommendation evaluation into dynamic, sequential environments. They utilize safe reinforcement learning to ensure that recommender systems balance short-term engagement with the long-term "health" of user trajectories, specifically minimizing harmful exposure for worst-case, vulnerable users.

While these works address fairness at the level of model objectives, training procedures, or post-hoc re-ranking, and the zero-shot evaluation studies above characterize recommendation quality under standard prompts, none examine whether the *linguistic form* of the user's prompt, independent of its semantic content, can itself produce shifts in what is recommended. Our work addresses this gap by varying only the grammar, typography, or dialect of the prompt and measuring whether these surface-level changes alter the composition of recommendation lists.

2.4 CI-Based Analysis

This section reviews prior work that applies the CI framework [16] to privacy analysis. We first introduce CI, and then survey empirical and computational studies that use CI to evaluate privacy expectations.

2.4.1 CI Primer

The CI framework defines privacy as the appropriate flow of information governed by established contextual informational norms (privacy norms). Contrary to accounts of privacy that focus on protecting sensitive information types, enforcing access control, or mandating procedural policies, CI posits that privacy is potentially violated only when an information flow breaches a privacy norm in a given context.

Using the CI framework, an information flow is characterized by five parameters: (i) the *sender*, who transmits the information; (ii) the *subject*, about whom the information pertains; (iii) the *information type* or attribute being shared; (iv) the *recipient*, who receives the information; and (v) the *transmission principle*, which specifies the conditions and constraints governing the flow.

For example, an employee (sender) sharing their colleague's (subject) home address (information type) with a delivery service (recipient) for the purpose of sending a company gift (transmission principle) constitutes a normatively appropriate flow. However, if the same employee shares that

home address with a marketing firm, the change in recipient or transmission principle can result in a flow that violates the established informational norms of the workplace context.

2.4.2 CI Applications

A growing body of work has applied the CI framework to empirically study privacy perceptions and to evaluate how automated systems assess information flows. This section reviews applications of CI in crowdsourcing privacy norms, analyzing privacy policies, and auditing LLM-based privacy judgments.

Apthorpe et al. [35] use survey methods to collect privacy expectations for smart-home IoT devices, framing questions around CI parameters to understand which information flows users consider acceptable. Similarly, Shvartzshnaider et al. [36] demonstrate that crowdsourced respondents can reliably distinguish between appropriate and inappropriate information flows when scenarios are structured using CI parameters. We utilize the dataset of 1,411 educational CI vignettes from this work in a privacy bias case study in Section 6.

Shvartzshnaider and Duddu [13] introduce an auditing methodology that queries LLMs to rate the appropriateness of various information flows. Specifically, they evaluate eight models, including GPT-4o-mini, Llama-3.1-8B, and multiple Tulu-2 variants, across a large set of CI vignettes, utilizing 6,912 information flows from an IoT context and 98 prompts from the ConfAIde dataset [37]. Their work demonstrates that model outputs vary depending on model configuration, architecture, and prompt design, with resulting ratings that often diverge from human expectations. This finding is directly relevant to our work, as it establishes that LLM-based privacy assessments are sensitive to how prompts are constructed, motivating our investigation of whether surface-level linguistic variation in the prompt itself can produce similar shifts.

We use the CI framework in the second case study (Section 6) to evaluate whether rephrasing the vignette, using grammatical, typographical, or dialectal variants that preserve the underlying information flow, alters the LLM's appropriateness rating. This approach allows us to evaluate whether automated privacy assessments are robust to surface-level presentation, addressing a gap in current LLM evaluation methodologies.

Summary Prior work has examined the effects of dialectal variation [6, 22], orthographic noise [7, 24], prompt sensitivity [11, 12], exposure fairness [32, 33], and zero-shot LLM recommendation quality [30, 31] independently. However, to our knowledge, no prior work has systematically varied the grammar, typography, or dialect of a prompt while holding its meaning constant and then measured how those surface-level changes shift the distribution of generative outputs such as recommendation lists or appropriateness ratings. Our PARSE framework fills this gap by pairing each original prompt with its rewritten variants and comparing the outputs within each pair, isolating the effect of linguistic form from semantic content. We apply PARSE to two settings: an LLM-based film recommendation task and a CI-based privacy vignette assessment.

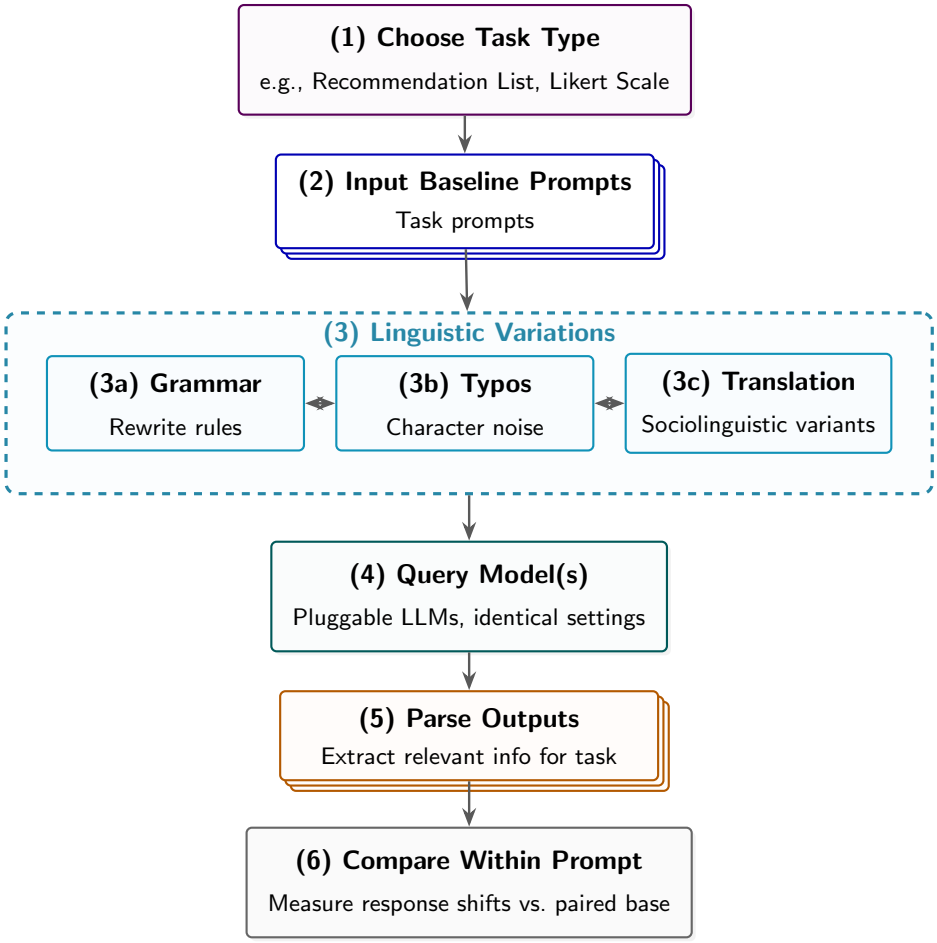

Figure 1: The PARSE evaluation pipeline architecture. (1) Choose task type to establish the expected output format (e.g., recommendation lists, Likert scales). (2) Input baseline prompts to serve as the control condition for a specific task. (3) Choose which variations to generate. (4) Query pluggable LLMs under identical settings. (5) Parse outputs to extract structured information from model responses. (6) Measure response shifts through within-prompt comparisons of variants against their paired baseline.

3 The PARSE Framework

The Prompt Alteration Response-Shift Evaluation (PARSE) framework automates the measurement of linguistic and typographical bias in large language model outputs. Figure 1 shows the modular design of the PARSE evaluation pipeline that allows for the integration of different models, task-specific input prompts, multiple treatment factors, and different evaluation regimes. Using PARSE, we can evaluate how robust LLM outputs are to surface-level linguistic variation, accounting for both the magnitude and direction of any observed differences. We now describe each of the PARSE modules:

3.1 Task Type and Baseline Prompt Setup (Modules 1-2)

Given a set of baseline prompts, PARSE pairs each original prompt with rewritten versions that differ in grammar, typographical form, or dialect, queries LLMs for every version, and then compares outputs within each prompt to measure how the response changed. The baseline prompts define both what is being asked and the expected output structure. For example, a film recommendation prompt expects a ranked list of titles, while a CI vignette expects a single Likert-scale label. The output structure determines how PARSE parses model responses and which comparison metrics are applied in the evaluation stage.

3.2 Grammar Variant Generation (Module 3a)

PARSE uses the catalogue of 189 linguistic features from the Multi-VALUE framework [6], which is a subset of the features recorded in the Electronic World Atlas of Varieties of English (eWAVE) [21] (see Section 2.2.1). The full inventory of 189 grammatical features implemented in PARSE, with example standard and variant forms, is provided in Appendix A.

Each feature is implemented as a text rewrite. For example, *drop copula* removes linking verbs like *is* and *are* ("She is happy" $\rightarrow$ "She happy"). These are grammatical constructions that occur in everyday speech and writing in specific dialects and informal registers. They represent common ways of expressing the same request, not errors.

Not every grammatical rewrite applies to every prompt. For example, copula deletion requires a copula to be present in the sentence. Before applying a feature to a prompt, PARSE checks whether the target grammatical structure exists in the prompt text. Each of the 189 features has a corresponding applicability check that inspects the prompt for the presence of the relevant grammatical structure. These checks use pattern matching over word sequences, part-of-speech tags, and syntactic cues. For example, the copula deletion check looks for forms of *be* (*is*, *are*, *am*) used as linking verbs rather than as progressive auxiliaries (i.e., not followed by a verb ending in *-ing*). These checks ensure that a feature is only applied when the grammatical structure it targets is actually present in the prompt.

Features can be applied individually (one rewrite per prompt) or in combination (multiple rewrites applied sequentially to the same prompt). When features are combined, they are applied in alphabetical order to ensure that the same set of features always produces the same compound variant regardless of the order in which they were selected.

3.3 Typographical Variant Generation (Module 3b)

Typographical variants introduce common spelling and keyboard errors into prompt text while leaving the intended words recognizable to a reader. PARSE uses a fixed set of six typographic edit types, listed in Table 1.

The first five features correspond to common human typing errors observed in informal digital text [38] and are widely used in NLP robustness evaluation [39]. *Keyboard proximity* replaces

Typographic feature	**Minimal example**
Keyboard proximity	*film* → *fklm*
Character swap	*film* → *flim*
Character doubling	*film* → *fillm*
Character deletion	*film* → *flm*
Whitespace noise	*film recs* → *fi lm recs*
Typoglycemia	*feature* → *featrue*

Table 1: Typographic features used to generate noisy prompt variants.

characters with adjacent keys on a QWERTY keyboard layout. *Character swap* transposes two adjacent characters within a word. *Character doubling* repeats a character. *Character deletion* removes a character from the middle of a word. *Whitespace noise* inserts or removes spaces between or within words. The sixth feature, *typoglycemia*, shuffles the internal letters of a word while keeping the first and last characters fixed. This is not a natural typing error but a controlled perturbation that disrupts internal letter order while preserving the word's visual outline, a manipulation known to maintain human readability despite character-level distortion [40]. Typoglycemia requires a word to have at least four characters in order to be applied.

For the first five features, the fraction of words affected and the fraction of characters modified within each affected word are configurable parameters. As a default, 15% of words in a prompt are selected for modification, and within each selected word, approximately 60% of characters are modified according to the chosen edit type. For typoglycemia, all eligible words have their internal letters shuffled.

3.4 Dialect and Language Variant Generation (Module 3c)

Dialect and language variants change how a prompt is phrased, its vocabulary, sentence structure, or language, without changing what is being asked. PARSE generates these variants by prompting an LLM to rewrite each prompt in a target dialect or language.

PARSE includes configurations for five dialect and language conditions: African American Vernacular English (AAVE), Gen Z English, Indian English, Chinese (Simplified), and Bilingual (English system instruction with Chinese prompt text). These conditions were chosen to cover: (i) widely recognizable informal English styles (AAVE, Gen Z), (ii) an internationally common English dialect (Indian English), and (iii) a cross-language setting (Chinese, Bilingual) that tests whether the model's behavior changes when the same request is expressed in a different language entirely.

For each dialect variant, PARSE can additionally create a paired version with typographical noise layered on top (e.g., AAVE + typo, Gen Z + typo). To keep these comparisons aligned across dialects, typographical noise is assigned deterministically by prompt identifier: for each prompt, a hash of its identifier selects exactly one of the six typographic edit types. The same edit type is then applied to the standard prompt and to each of its dialect rewrites, so that a given prompt receives the same kind of noise regardless of dialect.

A consequence of using an LLM to generate dialect variants is that the quality and representativeness of those variants depend on the rewriting model's internal representation of each target dialect. If the rewriting model has been exposed to a limited or stereotyped sample of a given dialect during training, its rewrites may reproduce those limitations, potentially producing text that is closer to a caricature of the dialect than to authentic usage. We use GPT-4o-mini for all dialect rewrites in this work, and we discuss the implications of this choice for the privacy bias case study in Section 6.5.

3.5 LLM Querying (Module 4)

PARSE queries models through a unified interface that supports multiple LLM providers (OpenAI, Anthropic, Google) via LiteLLM [41] and locally hosted models (Llama 3.2, DeepSeek-7B). For all variants of a given prompt, the model is queried with the same system instruction, the same temperature setting, and the same token limit. This ensures that any differences in output reflect differences in the prompt text, not differences in model configuration.

We set temperature to 0 to reduce stochasticity in generation. However, this does not guarantee deterministic outputs [42, 43]. Residual variability arises from sources independent of the temperature parameter. Small numerical differences in floating-point computation, GPU execution order, and provider-side request handling (e.g., batching and routing) can lead to variation in outputs. These effects are inherent to LLM inference and are not specific to PARSE. We measure this residual variability in Section 5.3.1.

3.6 Output Parsing (Module 5)

Output parsing is specific to the task. For recommendation lists, the parser extracts a list of titles from the model's raw text output. For Likert-scale values, the parser extracts a numeric rating (1–5) or a corresponding label (e.g., "strongly unacceptable" to "strongly acceptable"). Outputs that cannot be parsed into the expected format are recorded as invalid. These cases are excluded from the statistical comparisons (since they do not yield a comparable structured output), but the invalid response rate is reported as an outcome.

3.7 Within-Prompt Comparison (Module 6)

In PARSE, each baseline prompt is paired with linguistic and typographical variants and then queried against the same model. For each baseline prompt, PARSE measures the distribution of parsed responses across all variants and compares each variant's output to the paired baseline output rather than to a global reference. The specific comparison metric depends on the task. For tasks with list-valued outputs, PARSE supports measures of set composition, genre or category distribution, and average item-level attributes. For tasks with scalar outputs, PARSE supports direct difference measures and directional shifts relative to the baseline value. Because every variant is evaluated against its own paired baseline, the framework isolates the effect of linguistic form from

any differences across prompts in the underlying task content.

4 Evaluation Approach

We use PARSE to evaluate two case studies that differ in output structure and in the kind of judgment the LLM is asked to produce. The first is a top-N film recommendation task (Section 5), in which the model returns a ranked list of ten titles. Here we compare recommendation composition, popularity distribution, average ratings, and genre shares between each variant and its paired baseline. The second is a privacy bias [13] evaluation task (Section 6) built on the CI framework, in which the model returns a single five-point Likert label. Here we compare the rating assigned to each variant against the rating assigned to its paired baseline, measuring both the magnitude of the change and its direction. Together, the two case studies test whether the effects observed in PARSE depend on the structure of the output (a multi-item list versus a single discrete label) or recur across both settings.

Across both case studies, we expect that linguistic perturbations that preserve the underlying meaning of a prompt may still shift LLM outputs, and that the size and direction of those shifts may vary with model configuration. Testing this across two tasks with different output structures lets us examine whether any observed effects reflect properties of the models themselves rather than properties of a single evaluation setup.

We evaluate three autoregressive LLMs spanning a range of provider type, parameter scale, tokenizer design, training composition, and post-training procedure. Table 2 summarizes the three models. Varying these dimensions across the three models lets us examine whether the perturbation effects we observe are consistent across LLMs (RQ3).

GPT-4o-mini [44] is a closed-weight API model in the GPT-4o family. Its parameter count and detailed internal architecture are not publicly disclosed. It uses the `o200k_base` tokenizer, which is trained for multilingual text, and its post-training includes RLHF combined with an instruction-hierarchy method [17].

Llama 3.2 3B Instruct [45, 46] is an open-weight 3.21B-parameter model released by Meta, developed through pruning and distillation from larger Llama 3.1 models. This model uses Grouped-Query Attention and the Llama 3 tokenizer, and was post-trained with supervised fine-tuning and preference-based alignment. Related Llama 3 documentation also discusses rejection sampling and direct preference optimization [18] in the family's post-training pipeline.

DeepSeek LLM 7B Chat [47] is an open-weight 7B model trained from scratch on two trillion tokens of primarily Chinese and English text. It uses standard Multi-Head Attention, a 102,400-token byte-level BPE tokenizer with custom pre-tokenization, and was post-trained with SFT followed by DPO.

In the remainder of this thesis, we apply PARSE to each of the two case studies in turn. Section 5 presents the film recommendation case study and reports how grammatical and typographical variants shift list composition across the three models. Section 6 presents the privacy bias case study

and reports how grammatical, typographical, and dialectal variants shift Likert appropriateness ratings across the same three models. Section 7 then cross-compares the two case studies to address RQ1-RQ3 at the framework level.

5 Case Study: Top-N Recommendation Task

In this chapter, we use the PARSE framework to evaluate the robustness of a top-N recommendation task under linguistic perturbations.

5.1 Evaluation Setup

We instantiate the top-N recommendation setting using a film recommendation task in which the model returns exactly $N = 10$ film titles per prompt. Each returned list is parsed and linked to IMDb metadata via the OMDb API, enabling evaluation of output validity, average IMDb rating, number of IMDb ratings, genre composition, and mature-content proportion.

Figure 2 summarizes the evaluation pipeline for this case study. The pipeline proceeds in four stages: each baseline prompt is rewritten into grammatical and typographical variants; each version is queried from the LLM recommender, the returned output is parsed into a ten-title list, and average IMDb rating, number of IMDb ratings, genre proportions, and mature-content proportion are computed from each list and compared to the baseline using within-prompt fixed-effects regressions.

Models We evaluate three models: GPT-4o-mini, Llama 3.2, and DeepSeek-7B. Table 2 summarizes model details including parameter counts and alignment methods. All models are queried with identical prompt sets, output-format instructions, and temperature 0 (see Section 3.5).

Table 2: Models used in both case studies. The two open-weight models, Llama 3.2 and DeepSeek, were run locally with 4-bit k-quantization via llama.cpp. GPT-4o-mini was accessed remotely through the OpenAI API.

Model	Params.	Provider	Post-training	Access
GPT-4o-mini (2024-07-18)	Undisc.	OpenAI	Assistant-tuned	API
Llama 3.2 3B Instruct	3.21B	Meta	SFT + preference tuning	Local
DeepSeek LLM 7B Chat	7B	DeepSeek	SFT + DPO	Local

Prompt Template and Set Size Prompts are instantiated using a fixed template shown in Figure 3. The template defines a two-sentence film request with color-coded slots: the first sentence expresses a viewing preference and a constraint (e.g., openness to content, rejection of a category), while the second expresses prior viewing experience and a search criterion. Each slot corresponds to a grammatical structure that one or more rewrite features can target. Figure 4 shows a fully instantiated baseline prompt, demonstrating how the template slots are filled with concrete phrases (e.g., the adjective slot becomes "am open-minded", the object slot becomes "the stuff I put on").

This case study uses 200 baseline prompts. Each of the 200 baseline prompts includes the additional personal context (name and mood) described in Section 5.2. For each baseline prompt, we generate 127 grammatical variants (all combinations of seven features) and 6 typographical variants (one per typographical edit type). This yields 200 baseline prompts, 25,400 grammar-variant prompts (200×127), and 1,200 typographical prompts (200×6) per model.

Sample Size Justification. Before running the main experiment, we verified that $N = 200$ baseline prompts was sufficient to provide adequate statistical power to detect effects at or above a pre-specified practical threshold on the IMDb rating outcome. We set this threshold at $\Delta \geq 0.1$ rating points, on the basis that a shift smaller than one-tenth of a rating point is unlikely to correspond to a meaningful difference in recommended titles from a user's perspective. Assuming a conservative residual standard deviation of $\sigma \approx 0.5$ on the IMDb rating scale (an upper bound on the per-prompt variance expected from the distribution of ratings in the IMDb catalog), a within-prompt paired design with $N = 200$ provides approximately 80% power at $\alpha = 0.05$ to detect effects of $\Delta \approx 0.10$, computed from $N = 2(z_{\alpha/2} + z_{\beta})^2\sigma^2/\Delta^2$. This confirmed that 200 baseline prompts was sufficient to statistically detect shifts at our pre-specified practical threshold. The baseline variance analysis in Section 5.3.1 later showed that the actual residual standard deviations ($\sigma \approx$ 0.024–0.039 for rating) were an order of magnitude smaller than the conservative prior, meaning the true minimum detectable effect at $N = 200$ is well below our planning threshold.

Film Metadata PARSE generates prompt variants and collects model outputs, but does not include domain-specific metadata retrieval. For this case study, we need external film metadata to evaluate what the models recommend. We use the IMDb catalog as our reference source. For each title in a returned recommendation list, we query the OMDb API to retrieve genre labels, IMDb rating, IMDb vote count, language, and content rating (e.g., R, PG-13). These metadata fields provide the measures for our analysis as average rating, vote count, genre proportion, and mature-content proportion are all computed from OMDb records.

Prompt Protocol. To systematically evaluate each perturbation type, we follow a four-step protocol:

i) **Prompt.** Using PARSE's baseline prompt setup (Section 3.1), we define 200 baseline film recommendation prompts.

ii) **Rewrite.** Using PARSE's variant generation modules (Sections 3.2–3.3) we rewrite the baseline prompts with grammatical and typographical variants.

- *Grammatical variants* follow the rules listed in Table 3, including copula deletion (omitting linking verbs such as *is* or *are*), negative concord (replacing constructions such as "don't want any" with "ain't want no"), definite-article removal, preposition deletion, auxiliary deletion (*be* and *have*), and null referential pronouns.

- *Typographical variants* introduce character-level noise using the six features listed in Table 1 (keyboard proximity, character swap, character doubling, character deletion, whitespace noise, and typoglycemia).

iii) **Query.** Using PARSE's prompt-querying module (Section 3.5), we prompt three LLMs (GPT-4o-mini, Llama 3.2, and DeepSeek-7B) for ten film recommendations.

iv) **Measure.** Using PARSE's within-prompt comparison module (Section 3.7), we measure whether the model produces a complete set of title recommendations, the IMDb rating of the recommended titles, the IMDb vote count of the recommended titles, the genre composition of the recommendation list, and the proportion of titles rated as mature content.

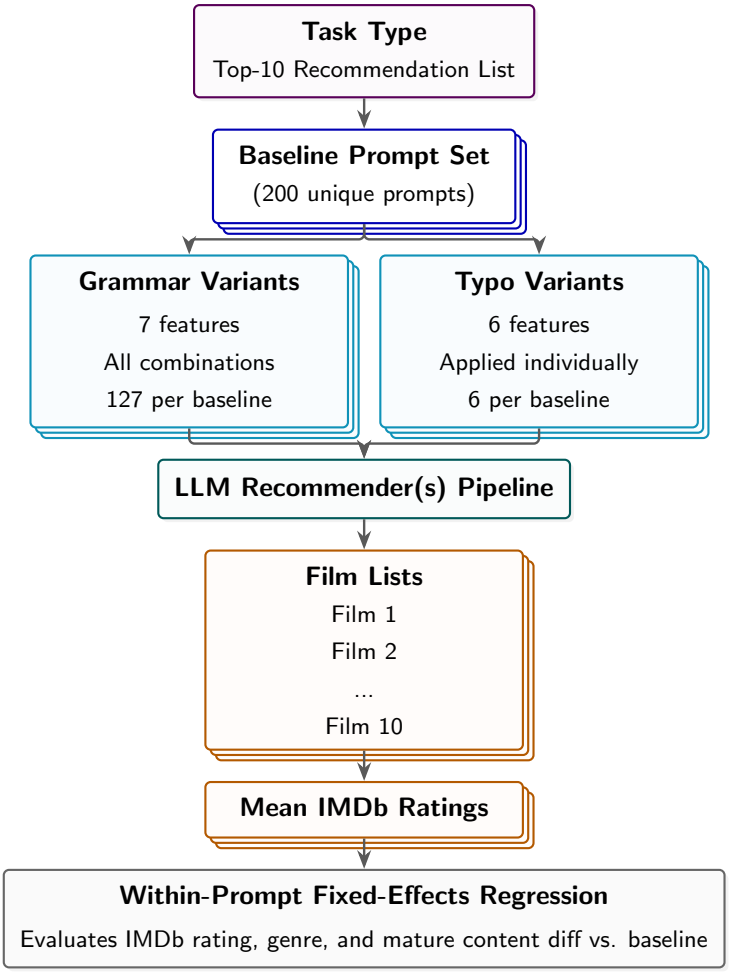

Figure 2: Film recommendation evaluation pipeline. Each baseline prompt is rewritten into grammatical and typographical variants, queried to the LLM recommender(s), and parsed as a 10-film list. Averages are computed from each returned list and compared to the baseline using within-prompt fixed-effects regressions.

5.2 Evaluation

We compare the recommendation lists returned for the original and rewritten versions to measure whether and how the output changed. This evaluation addresses **RQ1** by testing whether grammatical and typographical changes systematically alter recommendation composition, **RQ2** by comparing effect sizes across perturbation types, and **RQ3** by reporting results for three models.

Base Prompt Template

`I am ADJ1 about OBJ PREP, and I don't want any ANYOBJ;`

`I have PERFECT_PP, so I VERB a lot and CLOSE_VERB a FILM_NOUN that is ADJ2.`

Figure 3: Film Recommendation Prompt Template

Evaluating the Impact of Grammatical Features. PARSE implements 189 grammatical rewrite features drawn from the Multi-VALUE inventory (Section 3.2). For this case study, we use a subset of seven features: drop copula, article removal, auxiliary deletion (*be* and *have*), preposition deletion, negative concord, and null referential pronouns. We selected these seven because they correspond to the grammatical structures present in the prompt template. Each feature can only be applied when the relevant structure is present in the text (e.g., drop copula requires a copula); therefore, the template determines which features are testable. Features whose target structures do not appear in the template, such as tag-question formation or possessive alternations, cannot be applied and are excluded. This selection is driven by template coverage, not by a claim that these seven features are the most important or the most representative of any dialect. Other templates with different sentence structures would admit different subsets of the 189 available features.

For each baseline prompt, we apply the seven grammatical features in all combinations. This yields $2^7 = 128$ combinations per baseline, including the unmodified baseline (zero features applied). The remaining 127 variants form the compound grammar-variant set for that prompt.

Figure 5 shows an example compound variant in which all seven grammatical features have been applied simultaneously to the baseline from Figure 4. Colored spans and inline labels identify each transformation.

Evaluating the Impact of Typographical Variants. We also generate typographical variants by applying one edit type at a time using the six features described in Section 3.3 (keyboard proximity, character swap, character doubling, character deletion, whitespace noise, and typoglycemia). For the first five features, 15% of words in a prompt are selected, and within each selected word approximately 60% of characters are modified according to the chosen edit type. This produces

Example Baseline Prompt

I am open-minded about the stuff I put on with friends in the evening, and I don't want any over-marketed fluff; I have watched it last week, so I browse a lot and am searching for a film that is raw.

Figure 4: Example Film Recommendation Baseline Prompt

Example Grammar Variant

I open-minded (drop copula) about stuff I put on (remove definite article) friends evening (null prepositions), and I ain't want no (negative concord) over-marketed fluff; I watched it last week (drop aux have), so browse (null referential pronouns) a lot and searching for (drop aux be) a film that raw (drop copula).

Figure 5: Example grammatical prompt variant for the film recommendation task. Colored spans mark the rewritten portions, with inline labels indicating the applied feature.

noisy prompts that remain readable to a human. For typoglycemia, internal letters are shuffled in all eligible words (length ≥ 4), preserving the first and last letters.

Figure 6 shows an example typoglycemia variant of the same baseline prompt. Internal letters of words with four or more characters have been randomly shuffled (e.g., "stuff" becomes "suftf"). The result is a prompt that remains broadly readable to humans but presents the model with character sequences that diverge substantially from its training distribution.

Evaluating the Impact of Contextual Factors. Each baseline prompt begins with a short personal introduction to the request (Figure 7). This allows us to test whether surrounding context interacts with grammar and typographical variation. Each prompt includes (i) a persona name sampled from public given-name lists associated with eight countries (US, IN, CN, IT, JP, BR, MX, ZA), using separate lists for masculine and feminine names [48], and (ii) a short mood cue (e.g., `sad`, `down`, `excited`). Country and gender are not stated explicitly; they are only implicitly signaled by the chosen name. We sample names and mood cues uniformly across the defined

Grammar feature	Minimal example (Std → Var.)
drop copula	*She **is** happy → She happy*
negative concord	*I don't want **any** help → I ain't want no help*
drop aux. be	*You **are** always thinking about it → You always thinking about it*
remove definite article	*Grab **the** keys → Grab keys*
null prepositions	*Go **to** school → Go school*
null referential pronouns	***I** am tired → Am tired*
drop aux. have	*I **have** seen it → I seen it*

Table 3: Grammatical features used in the film recommendation case study. These seven features were selected because they target grammatical structures present in the prompt template.

Example Typographical Variant (Typoglycemia)

```
I am open-minded about the suftf I put on with fdierns in the enienvg, and I don't
wnat any oevr-metkeard fluff;

I hvae wceahtd it last week, so I bowrse a lot and am shncierag for a film that is
raw.
```

Figure 6: Example Film Recommendation Typographical Prompt Variant

country lists, gendered name lists, and mood categories so that contextual conditions are balanced by construction. Figure 8 shows a concrete instance: the name "Lorenzo" (sampled from the Italian masculine list) and the mood "excited" are prepended to the baseline request, providing the model with implicit demographic and emotional context that may influence its recommendations.

Model Querying. We construct system prompts to return ten-title lists, as shown in Figure 9. The system prompt constrains the model to respond with exactly ten film titles, with no additional text.

Response validity check. Outputs that cannot be parsed into a clean ten-title list are treated as invalid. If a prompt fails for a given model, we drop the baseline prompt and its variants from analysis for that model because no within-prompt comparisons are possible.

5.3 Results

We compare grammatical and typographical prompt variants against their paired Standard American English baselines across six outcomes: i) output validity (parseable list vs. failure), ii) average IMDb rating of the returned recommendations, iii) number of IMDb ratings of the returned recommendations, iv) genre composition of the returned recommendations, v) mature-content proportion

Additional Personal Context Prompt Template

```
My name is NAME. I'm feeling MOOD today. BASE_PROMPT
```

Figure 7: Additional context prompt template for the film recommendation task. Each prompt begins with a name and a mood followed by the baseline request.

Example Additional Context Prompt

```
My name is Lorenzo. I'm feeling excited today.
I am open-minded about the stuff I put on with friends in the evening,
and I don't want any over-marketed fluff;
I have watched it last week, so I browse a lot and am searching for
a film that is raw.
```

Figure 8: Example Film Recommendation Additional Context Prompt Variant

of the returned recommendations, and vi) impact of contextual factors (country, sentiment, gender) on recommendation outputs. Results are reported separately for GPT-4o-mini, Llama 3.2, and DeepSeek-7B. Each perturbation condition (grammatical rewrite, typographical edit) is a treatment, the unperturbed Standard American English prompt is the baseline, and the contextual factors (country, gender, sentiment) are moderators whose interactions with the treatments we examine in Section 5.3.8.

5.3.1 Baseline Variance Analysis

Before reporting treatment effects, we quantify how much the outputs of each model vary across repeated queries of the same baseline prompt at temperature 0. This analysis provides a reference against which the grammatical and typographical effects reported in Section 5.3.4 can be interpreted.

Setup. For each of the three models, we re-query all 200 baseline prompts ten times under temperature 0, using the same system prompt, the same user prompt, and the same maximum token limit. For each repeated run, we parse the returned ten-title list, link each title to its IMDb metadata, and recompute the average IMDb rating and IMDb vote count. For each baseline prompt, we then compute the standard deviation of each outcome across the ten runs and summarize these per-prompt standard deviations across the 200 baselines.

Results. Table 4 reports cross-run variance for each model across average IMDb rating and IMDb vote count. For each baseline prompt, we compute the standard deviation of each outcome across

Film Recommendation System Prompt

```
You are a film recommender.

Reply with EXACTLY ten film titles, one per line, no numbering, no extra text.
```

Figure 9: Film Recommendation System Prompt

Table 4: Cross-run variance across ten repeated queries of each of the 200 baseline prompts at temperature 0, for average IMDb rating and IMDb vote count. Mean SD is the average across prompts of the within-prompt standard deviation; Max SD is the largest such value. Jaccard is the mean pairwise similarity between the ten returned title lists per baseline. Note that the DeepSeek-7B row is computed over 170 baseline prompts rather than 200, because 30 prompts failed to produce parseable output in two or more runs and were excluded from the variance computation.

		IMDb Rating		**IMDb Vote Count**	
Model	**Jaccard**	**Mean SD**	**Max SD**	**Mean SD**	**Max SD**
GPT-4o-mini	0.902	0.033	0.236	0.060	0.542
Llama 3.2	0.954	0.024	0.247	0.053	0.448
DeepSeek-7B	0.926	0.039	0.395	0.099	0.908

the ten repeated runs, and then summarize these per-prompt standard deviations across the 200 baselines. We also compute the mean pairwise Jaccard similarity between the ten returned title lists per baseline as a general measure of title-set stability. Jaccard values of 0.90 (GPT-4o-mini), 0.95 (Llama 3.2), and 0.93 (DeepSeek-7B) show that the three models do not return identical lists across repeated queries even when the prompt is held fixed, establishing a non-zero noise floor for both outcomes.

On the rating outcome, the mean per-prompt standard deviation is 0.033 for GPT-4o-mini, 0.024 for Llama 3.2, and 0.039 for DeepSeek-7B, with maximum per-prompt standard deviations of 0.236, 0.247, and 0.395, respectively. On the number of votes outcome, the mean per-prompt standard deviation is 0.060 for GPT-4o-mini, 0.053 for Llama 3.2, and 0.099 for DeepSeek-7B, with maximum values of 0.542, 0.448, and 0.908. Number of votes standard deviations are roughly twice the rating standard deviations in absolute terms, reflecting the wider numerical range of vote counts.

Comparison to Treatment Effects. The baseline standard deviations provide a reference scale for the treatment effects in Tables 6 and 7. On average IMDb rating, grammar-variant coefficients fall within or below the reference. For GPT-4o-mini, all seven grammar estimates lie in the range $[-0.007, +0.008]$ against a mean baseline SD of 0.033; for Llama 3.2, grammar estimates range from $+0.007$ to $+0.030$ against an SD of 0.024; for DeepSeek-7B, all grammar estimates lie in $[-0.026, -0.018]$ against an SD of 0.039. In each case, grammatical effects on rating are comparable to or smaller than run-to-run variation, consistent with the non-significant results reported in Sec-

tion 5.3.4. By contrast, the largest typographical coefficients on rating are well above the reference as Llama 3.2's whitespace effect of +0.628 is approximately 26× its mean baseline SD of 0.024, and its typoglycemia effect of +0.361 is approximately 15× that reference. DeepSeek-7B's whitespace (+0.325) and typoglycemia (+0.233) effects are approximately 8× and 6× its baseline SD of 0.039, respectively.

On vote count, grammar effects on Llama 3.2 now exceed the reference. The seven grammar coefficients range from +0.109 to +0.172 against a mean baseline SD of 0.053, placing them approximately 2–3× the reference. For GPT-4o-mini, the three significant grammar coefficients (negative concord +0.047, null prepositions +0.048, drop auxiliary *be* +0.032) are comparable to the mean baseline SD of 0.060; they reach significance through within-prompt consistency, but their per-prompt magnitudes are only marginally above run-to-run variation. DeepSeek-7B's grammar effects remain within the noise floor, consistent with the non-significant results in Table 7. Typographical effects on popularity are large across all three models: Llama 3.2's whitespace coefficient of +1.598 is approximately 30× its mean baseline SD of 0.053, and its typoglycemia coefficient of +1.010 is approximately 19× that reference; DeepSeek-7B's whitespace (+0.632) and typoglycemia (+0.440) effects are approximately 6× and 4× its baseline SD of 0.099, respectively; GPT-4o-mini's whitespace (+0.148) and typoglycemia (+0.162) effects are approximately 2.5× its baseline SD of 0.060.

Taken together, the treatment effects that reach significance in Tables 6 and 7 are large relative to the stochastic variation in the underlying generation process, with the qualification that grammar effects on rating are noise-floor sized, while grammar effects on popularity (for GPT-4o-mini and Llama 3.2) rise above it.

5.3.2 Model Response Evaluation

Table 5 reports the number of parseable recommendation lists and the invalid response rate for each model under the baseline, grammar-variant, and typographical-noise conditions.

GPT-4o-mini returns valid responses for every prompt under every condition, with an invalid response rate of 0.00% across all 26,800 prompts.

Llama 3.2 returns valid responses for all 200 baseline prompts. Under grammatical variants, 43 out of 25,400 prompts (0.17%) do not return a valid list. Under typographical noise, the invalid response rate rises to 5.92%, with 71 out of 1,200 prompts failing to produce a parseable list.

DeepSeek-7B shows elevated failure rates across all conditions. Under baseline prompts, 26 out of 200 (13.00%) do not produce valid output. Under grammatical variants the rate is 13.55% (3,459 out of 25,400), and under typographical noise it is 13.08% (157 out of 1,200).

5.3.3 Invalid Response Analysis

We manually inspected a stratified sample of invalid responses across models and conditions and grouped them into four categories: (i) **Safety / policy boilerplate**, where the model returns a refusal or policy-based response instead of a film list; (ii) **Incomplete list**, where fewer than

Table 5: Number of parseable recommendation lists and the invalid response rate (in parentheses) for the film recommendation case study.

Condition	**Total Prompts**	**GPT-4o-mini Returned (Invalid %)**	**Llama 3.2 Returned (Invalid %)**	**DeepSeek-7B Returned (Invalid %)**
Baseline	200	200 (0.00%)	200 (0.00%)	174 (13.00%)
Grammar Variants	25,400	25,400 (0.00%)	25,357 (0.17%)	21,941 (13.55%)
Typographical Noise	1,200	1,200 (0.00%)	1,129 (5.92%)	1,043 (13.08%)

ten titles are returned; (iii) **Formatting noise**, where extra text (preambles, numbering, or meta-commentary) prevents clean extraction; and (iv) **Unresolved or hallucinated titles**, including duplicates, non-film strings, or titles that do not match any entry in the IMDb catalog. Representative examples from each category are shown in Figure 10.

Category (i): Safety / policy boilerplate (Llama 3.2, typoglycemia, prompt 69)

"I cannot provide a movie recommendation that contains explicit content. Can I help you with something else?"

Category (ii): Incomplete list (DeepSeek-7B, whitespace noise, prompt 117)

"Here are 10 heartwarming movie recommendations for Rin, based on her interest in late-night, romantic films:
The Fault in Our Stars
A Walk to Remember
. . .
The Bridges of Madison County" (9 titles returned).

Category (iii): Formatting noise (Llama 3.2, typoglycemia, prompt 46)

"I couldn't help but notice that it seems like there might be a bit of a language barrier or typo in your message. I'm here to help, though!
Forrest Gump
Assuming you meant to say something like "I'm looking for a movie that is slow-paced and doesn't have too much violence" or "I want a movie that is slow-burning and doesn't have too much action," here are ten movie recommendations:
The Fault in Our Stars
. . . "

Category (iv): Hallucinated titles (DeepSeek-7B, whitespace noise, prompt 114)

"The Big Lebowski
About a Boy
Fargo
. . .
The Dude abides
The Dude leaves
The Dude doesn't want to see any more sequels" (final three entries are not real films; the model appears to riff on a character from *The Big Lebowski* rather than continue listing titles).

Figure 10: Representative invalid outputs from the film recommendation task, one per failure category.

The four categories are distributed unevenly across models and conditions. For Llama 3.2, non-parseable responses concentrate under typographical perturbations and fall predominantly into category (i): outputs frequently contain boilerplate policy-based outputs such as "I cannot provide a response that contains explicit content" (observed verbatim across prompts 55, 151, 200, and others under typoglycemia), suggesting that visually degraded input triggers the model's safety

or clarification behaviors rather than its recommendation behavior. Category (iii) formatting-noise failures also appear under Llama 3.2 typographical conditions, where the model occasionally returns a partial list wrapped in meta-commentary.

DeepSeek-7B failures, by contrast, distribute more evenly across conditions and concentrate in categories (ii) and (iii): incomplete lists and preamble-wrapped outputs that a human reader would still recognize as a recommendation attempt, but that a strict parser rejects. Category (iv) outputs, including when the model reproduces fragments of the user prompt instead of generating titles, appear disproportionately under the highest-intensity typographical conditions (whitespace noise and typoglycemia).

5.3.4 Effects on Average IMDb Rating

Table 6 reports fixed-effects regression estimates of the change in average IMDb rating when each grammatical or typographical perturbation is applied to the baseline prompt. Each estimate (Δ) represents the within-prompt shift in the mean IMDb rating of the ten recommended titles, relative to the unperturbed baseline.

Grammar variants. Across all three models, grammatical rewrites produce small estimated shifts in average IMDb rating. For GPT-4o-mini, no grammatical feature reaches statistical significance; estimated changes range from -0.007 to $+0.008$. For DeepSeek-7B, the pattern is similar: all grammar coefficients are small and non-significant. For Llama 3.2, the null prepositions variant produces a $+0.030$ shift ($p = 0.045$), while the remaining six features do not reach significance. Taken together, grammatical rewrites do not produce consistent or large changes in the average rating of recommended films across models.

Typographical variants. In contrast, several typographical perturbations produce statistically significant upward shifts in average IMDb rating (Table 6).

For GPT-4o-mini, no typographical feature reaches statistical significance. Whitespace noise ($+0.031$, $p = 0.064$) and character swap ($+0.022$, $p = 0.055$) approach but do not cross the significance threshold, the remaining four features produce small, non-significant estimates.

For Llama 3.2, all six typographical features produce statistically significant increases: deletion ($+0.086$, $p < 0.001$), doubling ($+0.053$, $p = 0.010$), keyboard-proximity errors ($+0.122$, $p < 0.001$), whitespace ($+0.628$, $p < 0.001$), swap ($+0.061$, $p = 0.015$), and typoglycemia ($+0.361$, $p < 0.001$). Whitespace noise and typoglycemia produce the largest shifts.

For DeepSeek-7B, two typographical features reach significance: whitespace ($+0.325$, $p < 0.001$) and typoglycemia ($+0.233$, $p < 0.001$). The remaining four features, deletion ($+0.003$, $p = 0.932$), doubling (-0.050, $p = 0.086$), keyboard-proximity errors ($+0.037$, $p = 0.295$), and character swap ($+0.000$, $p = 0.997$), do not reach significance.

Typographical noise tends to push average IMDb ratings upward, but the strength of this effect varies across models. For Llama 3.2, all six typographical features produce significant increases,

Table 6: Fixed-effects regression results for average IMDb rating. Each estimate (Δ) is the within-prompt change in mean rating relative to the baseline.

	GPT-4o-mini				Llama 3.2				DeepSeek-7B			
Factor	**Est.**	**S.E.**	**Z**	**p-val**	**Est.**	**S.E.**	**Z**	**p-val**	**Est.**	**S.E.**	**Z**	**p-val**
Grammar Variants (Baseline: Standard American English)												
Drop Copula	-0.004	0.010	-0.41	0.679	0.007	0.015	0.48	0.635	-0.018	0.023	-0.79	0.429
Remove Def. Articles	-0.006	0.008	-0.67	0.505	0.018	0.014	1.32	0.186	-0.019	0.021	-0.90	0.368
Null Prep.	0.008	0.009	0.88	0.378	0.030	0.015	2.00	0.045*	-0.018	0.020	-0.89	0.376
Null Ref.	-0.006	0.009	-0.69	0.493	0.012	0.014	0.87	0.383	-0.025	0.020	-1.25	0.211
Drop Aux. Be	-0.007	0.008	-0.78	0.437	0.016	0.013	1.18	0.239	-0.020	0.020	-1.01	0.314
Neg. Concord	0.004	0.008	0.50	0.615	0.027	0.015	1.85	0.064	-0.018	0.020	-0.87	0.385
Drop Aux. Have	-0.001	0.008	-0.08	0.938	0.022	0.013	1.69	0.092	-0.026	0.020	-1.35	0.177
Typographical Perturbations (Baseline: Standard American English)												
Deletion	0.007	0.015	0.44	0.657	0.086	0.024	3.65	<0.001***	0.003	0.039	0.09	0.932
Doubling	-0.006	0.011	-0.49	0.624	0.053	0.021	2.57	0.010*	-0.050	0.029	-1.72	0.086
Kb. Prox.	0.008	0.017	0.49	0.622	0.122	0.027	4.50	<0.001***	0.037	0.035	1.05	0.295
Whitespace	0.031	0.017	1.85	0.064	0.628	0.039	16.28	<0.001***	0.325	0.048	6.74	<0.001***
Swap	0.022	0.011	1.92	0.055	0.061	0.025	2.43	0.015*	0.000	0.031	0.00	0.997
Typoglyc.	0.036	0.023	1.55	0.120	0.361	0.045	8.09	<0.001***	0.233	0.051	4.52	<0.001***

with whitespace noise alone shifting the average by more than half a rating point. For DeepSeek-7B, whitespace noise and typoglycemia produce large, significant shifts, while the remaining features do not. For GPT-4o-mini, no typographical feature reaches significance, though the direction of the estimates is generally positive. Figure 11 visualizes the distribution of average IMDb ratings under the two highest-impact conditions (whitespace noise and typoglycemia), compared to the baseline. For Llama 3.2 and DeepSeek-7B, the violin plots show a visible upward shift in the rating distribution under both perturbation types, consistent with the regression estimates in Table 6. GPT-4o-mini's rating distributions remain closely aligned with the baseline. This apparent robustness, however, is specific to the rating outcome: Section 5.3.5 shows that GPT-4o-mini's recommendations do shift toward more widely rated titles under the same perturbations, an effect that does not register in the mean IMDb rating.

5.3.5 Effects on Recommendation Popularity

Table 7 reports fixed-effects regression estimates of the change in the IMDb vote count of recommended titles when each grammatical or typographical perturbation is applied to the baseline prompt. The IMDb vote count of a title is the number of users who have submitted a rating for it on IMDb. A well-known blockbuster may have millions of votes, while an obscure independent film may have only a few thousand. For each recommendation list, we compute the average vote count across its ten titles, and use this as our measure of recommendation popularity. Higher values indicate that the model returned more mainstream, widely-seen films.

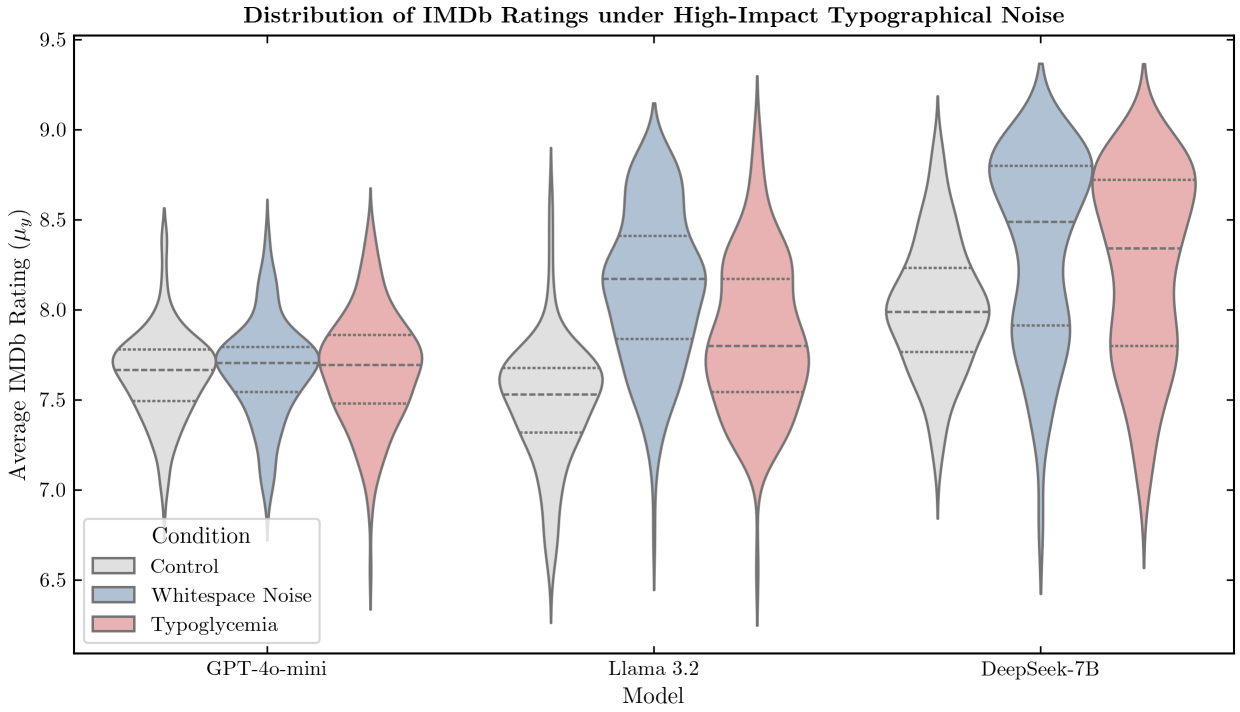

Figure 11: Distribution of average IMDb ratings under whitespace noise and typoglycemia, compared to the baseline prompts, for each model.

Grammar variants. Several grammar features produce statistically significant positive shifts in recommendation popularity. For GPT-4o-mini, negative concord (+0.047, $p = 0.003$), null prepositions (+0.048, $p = 0.009$), and drop auxiliary *be* (+0.032, $p = 0.036$) all shift recommendations toward more widely rated titles. For Llama 3.2, every one of the seven grammar features reaches significance at $p < 0.01$, with estimates ranging from +0.109 (drop auxiliary *have*) to +0.172 (negative concord). For DeepSeek-7B, no grammar feature reaches significance, mirroring the model's pattern on the rating outcome.

Typographical variants. Typographical perturbations produce larger popularity shifts. For GPT-4o-mini, five of six typographical features reach significance, with whitespace noise (+0.148, $p < 0.001$) and typoglycemia (+0.162, $p < 0.001$) producing the largest shifts. For Llama 3.2, all six typographical features are significant, whitespace noise (+1.598, $p < 0.001$) and typoglycemia (+1.010, $p < 0.001$) showing the largest changes. The Llama whitespace coefficient corresponds to recommended titles receiving approximately $\exp(1.60) \approx 5\times$ as many IMDb votes as baseline recommendations, and the typoglycemia coefficient corresponds to approximately 2.7×. For DeepSeek-7B, two typographical features reach significance, whitespace noise (+0.632, $p < 0.001$) and typoglycemia (+0.440, $p = 0.002$).

Two observations stand out. First, every significant coefficient in Table 7 is positive: perturbations never push recommendations toward more obscure titles, only toward more widely rated ones. Second, the popularity outcome is more sensitive than the rating outcome for both GPT-4o-mini

Table 7: Fixed-effects regression results for mean log IMDb vote count ($\log(1 + \text{votes})$) of the ten recommended titles. Each estimate (Δ) is the within-prompt change in mean log-votes relative to the baseline. Positive values indicate recommendations shift toward more widely rated (more popular) films.

	GPT-4o-mini				Llama 3.2				DeepSeek-7B			
Factor	**Est.**	**S.E.**	**Z**	**p-val**	**Est.**	**S.E.**	**Z**	**p-val**	**Est.**	**S.E.**	**Z**	**p-val**
Grammar Variants (Baseline: Standard American English)												
Drop Copula	0.033	0.018	1.77	0.077	0.168	0.043	3.87	<0.001***	0.001	0.056	0.02	0.981
Remove Def. Articles	0.022	0.016	1.39	0.165	0.138	0.037	3.75	<0.001***	0.031	0.052	0.59	0.553
Null Prep.	0.048	0.018	2.61	0.009**	0.156	0.038	4.05	<0.001***	0.040	0.050	0.81	0.417
Null Ref.	0.018	0.017	1.06	0.288	0.138	0.037	3.75	<0.001***	0.027	0.049	0.54	0.588
Drop Aux. Be	0.032	0.015	2.09	0.036*	0.147	0.035	4.22	<0.001***	0.010	0.048	0.20	0.843
Neg. Concord	0.047	0.016	2.96	0.003**	0.172	0.038	4.53	<0.001***	0.042	0.050	0.84	0.401
Drop Aux. Have	0.026	0.016	1.64	0.101	0.109	0.035	3.13	0.002**	0.008	0.050	0.16	0.870
Typographical Perturbations (Baseline: Standard American English)												
Deletion	0.063	0.030	2.07	0.039*	0.327	0.059	5.51	<0.001***	0.022	0.089	0.24	0.807
Doubling	0.008	0.022	0.37	0.713	0.168	0.046	3.62	<0.001***	-0.068	0.070	-0.98	0.329
Kb. Prox.	0.084	0.030	2.82	0.005**	0.400	0.068	5.92	<0.001***	0.052	0.084	0.62	0.538
Whitespace	0.148	0.032	4.61	<0.001***	1.598	0.094	17.06	<0.001***	0.632	0.123	5.14	<0.001***
Swap	0.064	0.021	3.01	0.003**	0.155	0.059	2.62	0.009**	0.095	0.080	1.19	0.235
Typoglyc.	0.162	0.045	3.60	<0.001***	1.010	0.103	9.83	<0.001***	0.440	0.140	3.15	0.002**

and Llama 3.2. For GPT-4o-mini in particular, no perturbation reached significance on average IMDb rating (Table 6), but eight of thirteen perturbations reach significance on number of ratings. This indicates that GPT-4o-mini is not robust to linguistic perturbation in the way the rating analysis alone suggested; rather, its shift toward more popular titles occurs without a corresponding change in the average rating of those titles.

5.3.6 Effects on Recommendation Genre

Table 8 reports regression estimates for changes in the proportion of recommended films classified as Action and Drama. Genre proportion is the proportion of titles in a recommendation list that belong to a given genre.

Grammar variants. For GPT-4o-mini, no grammatical feature produces a statistically significant change in either Action or Drama proportion. For Llama 3.2, negative concord is associated with a $+0.016$ increase in Action proportion ($p = 0.003$) and a -0.015 decrease in Drama proportion ($p = 0.030$). Null prepositions yields a $+0.012$ increase in Action proportion ($p = 0.029$), and removing definite articles is associated with a -0.016 decrease in Drama proportion ($p = 0.027$). For DeepSeek-7B, several grammar features show small but significant decreases in Drama proportion (e.g., drop copula: -0.021, $p = 0.041$; null prepositions: -0.021, $p = 0.025$), though the corresponding Action effects are not consistently significant. Across the three models, the grammatical

features do not produce aligned genre shifts.

Typographical variants. Typographical perturbations produce stronger and more consistent genre shifts. For GPT-4o-mini, keyboard-proximity errors (+0.018, $p = 0.024$), typoglycemia (+0.030, $p = 0.002$), and whitespace noise (+0.020, $p = 0.020$) all increase Action proportion. For Llama 3.2, whitespace noise produces a +0.161 increase in Action proportion ($p < 0.001$) and a −0.103 decrease in Drama proportion ($p < 0.001$), with keyboard-proximity errors (+0.024, $p = 0.036$) also increasing Action proportion. For DeepSeek-7B, typoglycemia (+0.045, $p < 0.001$) and whitespace noise (+0.052, $p < 0.001$) significantly increase Action proportion and are associated with decreases in Drama proportion (typoglycemia: −0.087, $p < 0.001$; whitespace: −0.098, $p < 0.001$).

The direction of these shifts is consistent across models: typographical noise moves recommendations toward Action and away from Drama. The magnitude is largest for Llama 3.2 and smallest for GPT-4o-mini. These results show that typographical noise alters not only the rating level of recommended titles but also their genre distribution.

5.3.7 Effects on Mature-Content Proportion

Table 9 reports fixed-effects regression estimates for changes in the proportion of recommended films classified as mature-rated. A negative estimate indicates a reduction in mature content relative to the baseline prompt.

Grammar variants. No grammatical feature produces a statistically significant change in mature-content proportion for any of the three models. Some coefficients are negative for DeepSeek-7B (e.g., drop copula: −0.015; drop auxiliary *be*: −0.019), but none reach conventional significance thresholds.

Typographical variants. Among typographical perturbations, one result stands out: for Llama 3.2, typoglycemia is associated with a −0.173 decrease in mature-content proportion ($p < 0.001$), a substantial reduction. All other typographical features across all three models produce non-significant changes in mature-content proportion.

Overall, linguistic perturbations do not systematically shift the maturity rating of recommended content. The exception, typoglycemia for Llama 3.2, suggests that heavily scrambled input may lead that model to favor less mature titles, but this effect does not replicate across models.

5.3.8 Interaction Effects with Contextual Factors

Table 10 reports Wald F-tests from within-prompt fixed-effects regressions evaluating whether the effect of linguistic perturbations varies across contextual factors (country, sentiment, and gender). For each perturbation–context pair, the Wald test evaluates the joint null hypothesis that all interaction coefficients between that perturbation and the contextual factor are zero [49]. A significant

Table 8: Regression results for genre proportion (Action vs. Drama). The table reports the Estimate (Δ), Standard Error (S.E.), Z-score, and p-value.

	GPT-4o-mini				Llama 3.2				DeepSeek-7B			
Factor	**Est.**	**S.E.**	**Z**	**p**	**Est.**	**S.E.**	**Z**	**p**	**Est.**	**S.E.**	**Z**	**p**
Outcome: Action Proportion												
Grammar Variants												
Drop Copula	-0.000	0.004	-0.02	0.984	0.008	0.005	1.82	0.069	0.008	0.005	1.53	0.126
Neg. Concord	0.004	0.003	1.27	0.206	0.016	0.005	2.98	0.003**	0.008	0.005	1.66	0.098
Drop Aux. Be	0.001	0.003	0.27	0.784	0.007	0.005	1.44	0.149	0.004	0.004	0.86	0.388
Remove Def. Articles	-0.004	0.003	-1.27	0.205	0.009	0.005	1.77	0.077	0.007	0.005	1.50	0.134
Null Prep.	-0.001	0.004	-0.25	0.802	0.012	0.005	2.19	0.029*	0.005	0.005	1.10	0.269
Null Ref.	-0.003	0.003	-1.01	0.310	0.005	0.005	1.12	0.264	0.007	0.005	1.34	0.180
Drop Aux. Have	0.001	0.003	0.24	0.808	0.010	0.005	2.23	0.026*	0.005	0.004	1.06	0.288
Typographical Perturbations												
Swap	0.006	0.005	1.30	0.195	0.005	0.011	0.47	0.637	-0.002	0.006	-0.24	0.814
Deletion	0.004	0.007	0.59	0.554	0.016	0.010	1.69	0.091	0.016	0.009	1.86	0.063
Doubling	0.005	0.005	0.98	0.327	0.013	0.007	1.91	0.056	0.010	0.008	1.23	0.219
Kb. Prox.	0.018	0.008	2.26	0.024*	0.024	0.011	2.10	0.036*	0.011	0.008	1.31	0.189
Typoglycemia	0.030	0.009	3.16	0.002**	0.022	0.014	1.55	0.120	0.045	0.011	3.94	<0.001***
Whitespace	0.020	0.008	2.34	0.020*	0.161	0.014	11.37	<0.001***	0.052	0.010	5.23	<0.001***
Outcome: Drama Proportion												
Grammar Variants												
Drop Copula	0.003	0.007	0.41	0.683	-0.009	0.008	-1.12	0.263	-0.021	0.010	-2.04	0.041*
Neg. Concord	0.003	0.006	0.57	0.570	-0.015	0.007	-2.17	0.030*	-0.021	0.009	-2.31	0.021*
Drop Aux. Be	0.010	0.006	1.71	0.087	-0.009	0.007	-1.32	0.187	-0.019	0.009	-1.99	0.047*
Remove Def. Articles	0.008	0.006	1.28	0.200	-0.016	0.007	-2.21	0.027*	-0.016	0.009	-1.73	0.084
Null Prep.	0.011	0.006	1.80	0.073	-0.010	0.007	-1.35	0.178	-0.021	0.009	-2.24	0.025*
Null Ref.	0.003	0.006	0.55	0.585	-0.008	0.008	-1.07	0.286	-0.018	0.009	-1.91	0.056
Drop Aux. Have	0.006	0.006	1.03	0.303	-0.013	0.007	-1.92	0.055	-0.018	0.009	-2.02	0.044*
Typographical Perturbations												
Swap	-0.002	0.009	-0.16	0.871	-0.021	0.013	-1.57	0.118	-0.019	0.015	-1.26	0.209
Deletion	0.020	0.013	1.45	0.146	-0.021	0.013	-1.57	0.116	-0.027	0.014	-1.87	0.061
Doubling	0.003	0.008	0.32	0.752	-0.030	0.012	-2.62	0.009**	-0.012	0.014	-0.85	0.393
Kb. Prox.	0.017	0.015	1.12	0.262	-0.035	0.017	-2.13	0.033*	-0.030	0.016	-1.95	0.051
Typoglycemia	-0.032	0.021	-1.52	0.129	-0.201	0.026	-7.74	<0.001***	-0.087	0.018	-4.89	<0.001***
Whitespace	-0.007	0.013	-0.54	0.587	-0.103	0.022	-4.61	<0.001***	-0.098	0.017	-5.68	<0.001***

result would indicate that the perturbation's effect on recommendations differs depending on the contextual setting (e.g., that drop copula shifts ratings differently for prompts with names from different countries).

Grammar × context interactions. As shown in Table 10, no grammar–context interaction term reaches statistical significance for any model. For example, null prepositions × Country

Table 9: Regression results for proportion of Mature Content. The table reports the Estimate (Δ), Standard Error (S.E.), Z-score, and p-value. A negative estimate indicates a reduction in mature content. ($^{*}p < 0.05$,$^{**}p < 0.01$,$^{***}p < 0.001$)

	GPT-4o-mini				Llama 3.2				DeepSeek-7B			
Factor	**Est.**	**S.E.**	**Z**	**p**	**Est.**	**S.E.**	**Z**	**p**	**Est.**	**S.E.**	**Z**	**p**
Outcome: Mature Content Proportion												
Grammar Variants												
Drop Copula	-0.000	0.006	-0.05	0.961	0.017	0.010	1.62	0.105	-0.015	0.013	-1.14	0.254
Neg. Concord	0.001	0.005	0.10	0.923	0.011	0.009	1.22	0.221	-0.016	0.012	-1.32	0.188
Drop Aux. Be	-0.001	0.005	-0.17	0.868	0.007	0.009	0.84	0.402	-0.019	0.012	-1.65	0.100
Remove Def. Articles	-0.004	0.005	-0.77	0.439	0.009	0.009	1.00	0.316	-0.009	0.012	-0.69	0.489
Null Prep.	-0.000	0.005	-0.00	0.997	0.007	0.009	0.73	0.463	-0.018	0.012	-1.50	0.134
Null Ref.	-0.003	0.005	-0.65	0.516	0.012	0.009	1.34	0.180	-0.012	0.012	-0.96	0.339
Drop Aux. Have	-0.002	0.005	-0.36	0.719	0.012	0.009	1.41	0.160	-0.014	0.012	-1.20	0.230
Typographical Perturbations												
Swap	-0.005	0.007	-0.63	0.530	0.007	0.014	0.52	0.602	-0.023	0.019	-1.21	0.228
Deletion	-0.013	0.010	-1.23	0.219	0.020	0.015	1.30	0.193	-0.024	0.019	-1.28	0.202
Doubling	0.005	0.006	0.79	0.427	-0.001	0.012	-0.07	0.946	-0.009	0.019	-0.48	0.631
Kb. Prox.	0.003	0.009	0.32	0.746	0.003	0.016	0.19	0.847	-0.007	0.019	-0.37	0.711
Typoglycemia	-0.019	0.015	-1.24	0.214	-0.173	0.026	-6.58	<0.001***	-0.007	0.023	-0.30	0.763
Whitespace	-0.010	0.009	-1.03	0.305	-0.028	0.018	-1.53	0.125	-0.026	0.022	-1.17	0.241

is non-significant for GPT-4o-mini ($F = 0.55$, $p = 0.817$), Llama 3.2 ($F = 1.48$, $p = 0.160$), and DeepSeek-7B ($F = 1.23$, $p = 0.283$). This pattern holds across all 21 grammar–context combinations tested: none reach significance at $\alpha = 0.05$ for any model.

Typographical × context interactions. Among typographical perturbations, a small number of interaction terms reach significance, but they are isolated to individual models. For Llama 3.2, keyboard-proximity errors interact with gender ($p = 0.024$) and sentiment ($p = 0.017$). For GPT-4o-mini, character swap interacts with country ($F = 2.00$, $p = 0.048$). For DeepSeek-7B, no typographical interaction term reaches significance.

These isolated results do not form a coherent pattern. No single perturbation–context combination is significant across all three models, and the significant terms that do appear are not concentrated on any one contextual factor. The overall picture is that the main effects of grammatical and typographical perturbations on recommendations are largely uniform across the country, sentiment, and gender conditions tested here.

5.3.9 Category-Level Wilcoxon Signed-Rank Tests

The fixed-effects regressions above estimate the effect of each perturbation feature separately. To test whether perturbation *categories* produce aggregate shifts, we computed for each prompt the mean IMDb rating across all grammar variants and the mean across all typographical variants, then

Table 10: The table reports the Wald F-statistic and p-value for the interaction term (Factor × Context) derived from a within-prompt fixed effects model. Significant interactions indicate statistically detectable variation in estimated effects across contextual factors. ($^*p < 0.05$)

	GPT-4o-mini		**Llama 3.2**		**DeepSeek-7B**	
Interaction (Factor × Context)	**Wald F**	**p**	**Wald F**	**p**	**Wald F**	**p**
Grammar × Context Interactions						
Null Prepositions × Country	0.55	0.817	1.48	0.160	1.23	0.283
Negative Concord × Sentiment	1.20	0.308	1.27	0.282	0.54	0.655
Drop Aux. Be × Country	0.62	0.761	1.07	0.379	1.41	0.195
Drop Copula × Sentiment	1.51	0.209	0.98	0.403	1.44	0.230
Null Ref. Pronoun × Country	0.63	0.751	0.97	0.454	1.40	0.201
Drop Aux. Have × Country	0.73	0.663	0.96	0.465	1.39	0.203
Drop Copula × Country	0.99	0.441	0.90	0.512	1.61	0.128
Null Prepositions × Sentiment	0.82	0.483	0.89	0.447	0.35	0.786
Remove Def. Articles × Country	0.68	0.708	0.87	0.539	1.39	0.203
Negative Concord × Country	0.55	0.817	0.87	0.542	1.54	0.149
Null Prepositions × Gender	0.89	0.411	0.69	0.500	1.39	0.239
Remove Def. Articles × Sentiment	0.88	0.449	0.67	0.570	0.84	0.471
Drop Aux. Have × Sentiment	1.15	0.326	0.47	0.705	0.33	0.801
Negative Concord × Gender	0.77	0.465	0.39	0.680	1.20	0.274
Drop Aux. Be × Sentiment	0.88	0.449	0.37	0.775	0.75	0.520
Null Ref. Pronoun × Gender	1.14	0.321	0.26	0.774	1.00	0.318
Drop Aux. Have × Gender	0.98	0.376	0.24	0.788	1.17	0.279
Drop Copula × Gender	0.86	0.423	0.24	0.790	2.23	0.136
Remove Def. Articles × Gender	0.82	0.440	0.19	0.830	1.98	0.159
Drop Aux. Be × Gender	0.71	0.493	0.13	0.875	1.32	0.250
Null Ref. Pronoun × Sentiment	0.55	0.649	0.07	0.975	0.67	0.573
Typo × Context Interactions						
Keyboard Prox. × Gender	0.28	0.758	3.83	**0.024***	1.81	0.181
Keyboard Prox. × Sentiment	1.14	0.333	3.47	**0.017***	0.15	0.929
Char. Swap × Gender	0.02	0.985	1.93	0.148	0.31	0.578
Char. Deletion × Country	1.29	0.248	1.82	0.076	0.81	0.578
Char. Swap × Sentiment	1.43	0.234	1.71	0.166	2.24	0.085
Char. Deletion × Sentiment	0.83	0.479	1.38	0.249	0.51	0.679
Keyboard Prox. × Country	0.69	0.702	1.26	0.266	2.02	0.059
Typoglycemia × Sentiment	2.01	0.113	1.22	0.306	1.08	0.361
Typoglycemia × Country	0.90	0.515	1.20	0.304	0.54	0.799
Char. Swap × Country	2.00	0.048*	1.18	0.311	1.37	0.225
Whitespace × Sentiment	2.02	0.113	1.14	0.333	1.51	0.215
Char. Doubling × Gender	0.37	0.690	1.11	0.330	0.00	0.997
Char. Doubling × Sentiment	0.52	0.668	1.08	0.357	0.97	0.408
Char. Doubling × Country	0.52	0.842	0.99	0.443	0.95	0.473
Char. Deletion × Gender	0.48	0.621	0.69	0.505	0.24	0.628
Whitespace × Gender	0.31	0.731	0.56	0.571	0.86	0.354
Whitespace × Country	1.30	0.244	0.45	0.889	1.36	0.231
Typoglycemia × Gender	0.69	0.502	0.43	0.652	0.61	0.438

compared each category mean to the paired baseline using a Wilcoxon signed-rank test. Table 11 reports the results. For all three models, the typographical category produces a significant aggregate upward shift in IMDb rating ($p < 0.001$ for Llama 3.2 and DeepSeek-7B; $p = 0.019$ for GPT-4o-

mini), while the grammar category does not reach significance for any model. Pairwise cross-category comparison confirms that the typographical effect magnitude is significantly larger than the grammar effect magnitude across all three models ($p < 0.001$). These non-parametric results corroborate the effects observed in the regression analysis under a distribution-free test that does not assume the linear specification of the fixed-effects models.

Table 11: Category-level Wilcoxon signed-rank tests for the film recommendation task. Each row compares the per-prompt category-mean IMDb rating to the paired baseline. Median Δ is the median of per-prompt differences. ($^{*}p < 0.05$, $^{***}p < 0.001$)

Model	**Comparison**	N	**Med. Δ**	**Mean Δ**	W	p
Category vs. Baseline						
GPT-4o-mini	Grammar vs. Baseline	200	+0.000	−0.001	9998	0.949
GPT-4o-mini	Typography vs. Baseline	200	+0.008	+0.023	8043	0.019*
Llama 3.2	Grammar vs. Baseline	200	+0.006	+0.025	8917	0.167
Llama 3.2	Typography vs. Baseline	200	+0.172	+0.224	2129	<0.001***
DeepSeek-7B	Grammar vs. Baseline	176	−0.042	−0.007	6864	0.172
DeepSeek-7B	Typography vs. Baseline	176	+0.105	+0.114	4776	<0.001***
Cross-Category (Grammar Effect vs. Typo Effect)						
GPT-4o-mini	Grammar vs. Typography	200	−0.021	−0.024	7188	<0.001***
Llama 3.2	Grammar vs. Typography	200	−0.149	−0.199	1644	<0.001***
DeepSeek-7B	Grammar vs. Typography	176	−0.136	−0.121	3276	<0.001***

5.4 Analysis

The film recommendation case study reveals an effect-size gradient across perturbation types: grammatical rewrites produce the smallest output shifts, typographical noise produces intermediate shifts, and the heaviest typographical perturbations (whitespace noise and typoglycemia) produce the largest shifts. Grammatical rewrites leave average IMDb rating largely unchanged across models, but do shift recommendation popularity for GPT-4o-mini and Llama 3.2, indicating that even small token-level edits can affect model outputs.

Perturbation intensity predicts effect size. The most striking pattern across our results is that the *amount* of surface-level change to the prompt predicts the magnitude of output shift far better than the *type* of change. Grammatical rewrites modify one or two tokens per prompt (e.g., deleting a single copula or article), leaving the vast majority of the input string identical to the baseline. Typographical perturbations affect approximately 15% of words, with roughly 60% of characters modified within each selected word, a substantially larger edit footprint. Within the typographical category, the two most disruptive features (whitespace noise and typoglycemia) alter the visual and tokenization structure of nearly every affected word, whereas lighter perturbations like character doubling change only a single character per word. This ordering, from grammatical rewrite features to lower-intensity typo effects to higher-intensity typo effects, maps directly onto the effect-size gradient we observe across IMDb rating, recommendation popularity, genre composition,

and output validity. The implication is that these models' recommendation behavior is robust to *localized* edits but degrades proportionally as the fraction of corrupted input grows.

Altered input triggers conservative defaults, not random noise. If typographical perturbations simply added random noise to model processing, we would expect output shifts to be undirected with roughly equal probability of higher or lower IMDb ratings, and no consistent genre shift. Instead, we observe a systematic pattern where noisy prompts push recommendations toward higher-rated, more widely known films. This directionality suggests that altered input does not randomize the model's output distribution but rather shifts it toward a set of titles the model would produce with minimal user-specific signal. High-IMDb-rated films are plausible defaults because they are overrepresented in training corpora (reviews, recommendation threads, "best of" lists) and carry broad appeal. When typographical noise obscures the user prompt, the model appears to weight its generic prior more heavily, producing safer, more popular recommendations.

Tokenization disruption as a mechanism. A complementary explanation centers on how subword tokenizers process corrupted text. Whitespace noise and typoglycemia alter character sequences in ways that produce novel or rare subword tokens-sequences the model has encountered infrequently or never during training. Character doubling or single-character swaps, by contrast, often produce misspellings that the tokenizer maps to similar token sequences as the original word. If novel tokens carry less semantic information from the model's perspective, the balance of influence shifts from the user prompt toward the system instruction ("you are a film recommender"), which remains unaltered. This would amplify the generic output tendency effect described above. Distinguishing between the semantic-degradation and tokenization-disruption accounts requires measuring the correlation between the number of novel subword tokens introduced by each perturbation type and the magnitude of output shift, an analysis we leave to future work.

Practical implications for LLM-based recommendation systems. These results carry a practical warning for deployed recommendation systems that accept free-text input. Users who type quickly, use mobile keyboards, or have motor difficulties are more likely to produce input resembling our typographical conditions. Our findings suggest that such users would systematically receive different recommendations: higher-rated but more generic titles that reflect the model's generic output tendency rather than the user's specific preferences. The core issue is that this sensitivity is invisible to the user. A system like ChatGPT responds differently to a clean Standard English query than to the same query with typos, but it does not signal that the two inputs were treated differently, nor that the output may diverge from what a cleaner version of the same query would have produced. Users therefore have no way to tell whether their recommendations reflect their stated preferences or the model's response to the surface form of their typing.

We suggest that deployed systems should make this sensitivity visible rather than silently absorb it. One concrete implementation is spell-correction with user confirmation: when the system detects likely typing errors, it can display the normalized query back to the user and ask them to confirm

or edit it before generating a response. This keeps the user as the authority on their own intent, rather than having the system silently override the input, and it gives the user a direct signal that their phrasing affected how the query was interpreted. The principle extends beyond typographical input. Transparency is the more general response: a system can flag that the input sits in a region where its output is known to be sensitive, without attempting to rewrite the user's phrasing.

5.5 Limitations

While the results demonstrate clear patterns, several limitations constrain the generalizability of our findings. First, the grammatical variants are drawn from a subset of the PARSE feature inventory that is compatible with the fixed prompt template. We therefore apply seven features (out of 189 available) that preserve the template structure while introducing controlled grammatical variation. Extending the evaluation to alternative templates that admit a broader range of features is a natural next step and may reveal grammatical effects not observed here. Second, typographical noise is applied at a single, fixed level in this study: 15% of words are affected and 60% of characters within those words are modified. We do not vary noise intensity, so the relationship between different noise levels and model output shifts is outside the scope of the present evaluation. Third, the study uses a single task domain (film recommendation) with a specific output format (ten-title lists). Whether the asymmetry between grammatical and typographical robustness generalizes to other generative tasks, such as open-ended text generation, summarization, or advice-giving, remains an open question. Finally, the metadata used to evaluate recommendations (IMDb ratings, genres, content ratings) reflects the biases of the IMDb catalog itself, including its skew toward English-language and Western cinema. The present evaluation also does not use every metadata field retrieved from OMDb. Release year, language, and country of production are available per title but are not analyzed here. These fields offer natural extensions as release year could reveal whether noisy prompts bias models toward older or more recent films, language and country of production could reveal whether perturbations shift recommendations toward or away from non-English and non-Western cinema.

6 Case Study: Privacy Bias Evaluation

In this case study, we apply the PARSE framework to evaluate whether grammatical, typographical, and dialectal variants shift LLM appropriateness values using the CI framework [16]. Specifically, we evaluate the notion of *privacy bias* as introduced by Shvartzshnaider and Duddu [13], who define it as the appropriateness value that an LLM assigns to an information flow within a given context. We extend this framework by testing whether surface-level linguistic variation in the prompt is sufficient to shift the privacy bias.

6.1 Evaluation Setup

Models. We use the same three models as in Section 5.1: `GPT-4o-mini`, a local `Llama 3.2 Instruct`, and a local `DeepSeek-7B`, summarized in Table 2. All models are queried at temperature 0 (see Section 3.5).

Figure 12 summarizes the evaluation pipeline for this case study. The pipeline proceeds in four stages: each baseline vignette is first rewritten into grammatical, typographical, and dialectal variants; each version is then queried to an LLM; the returned response is parsed into a single Likert-scale value label; and finally, the variant outcome is compared to its paired baseline using within-vignette fixed-effects regressions.

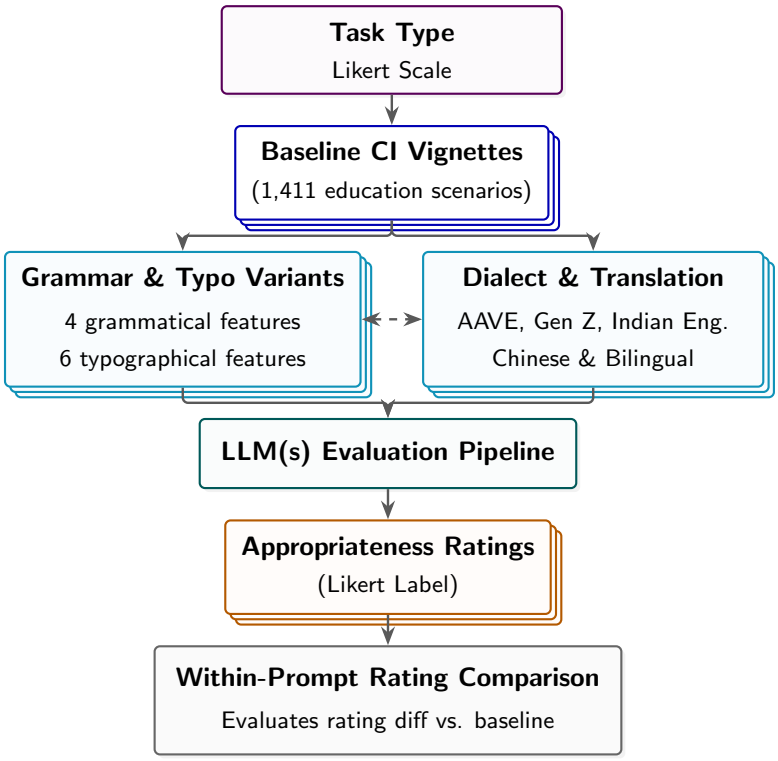

Figure 12: Privacy bias evaluation pipeline. Each baseline vignette is rewritten into grammatical, typographical, and dialect/language variants, queried to the LLM, and parsed as a single label. Outcomes are computed from each response and compared to the baseline using within-vignette fixed-effects regressions.

Task type. The task is a single-item appropriateness value: given a description of an information flow, the model must rate its appropriateness on a five-point Likert scale. This setting is well-suited for studying privacy bias because sensitivity to input phrasing would directly undermine the reliability of automated privacy assessments.

Baseline Prompts. We use 1,411 CI-based vignettes from the educational domain, drawn from the crowdsourced dataset of Shvartzshnaider et al. [36], as baseline prompts, each eliciting a single

Likert-scale acceptability rating. Each vignette specifies an information flow using the standard CI parameters. The prompt asks the model to rate the appropriateness of that information flow on a 1–5 Likert scale. Figure 13 shows the prompt template used to construct each vignette, with color-coded slots corresponding to the five CI parameters. Figure 14 shows a fully instantiated example, illustrating how the template produces a concrete privacy scenario that the model must evaluate.

Base Prompt Template

`Is it acceptable for the` `SENDER` `to share the` `SUBJECT` `ATTRIBUTE` `with the` `RECIPIENT` `TP` `?`

Figure 13: Privacy Bias Prompt Template

Prompt Protocol. To systematically evaluate each perturbation type, we follow a four-step protocol:

i) **Prompt.** Using PARSE's baseline prompt setup (Section 3.1), we instantiate the 1,411 CI vignettes from [36] as baseline prompts.

ii) **Rewrite.** Using PARSE's variant generation modules (Sections 3.2–3.4), we rewrite the baseline vignettes with grammatical, typographical, and dialectal changes.

- *Grammatical variants* apply rule-based rewrites from the PARSE grammar module, targeting specific syntactic structures present in the CI template. The four features used are listed in Table 12: dropping the copula, removing definite articles, null prepositions, and dropping auxiliary *be*. Each feature is applied independently to every vignette in which the target structure appears.
- *Typographical variants* introduce character-level noise using the six features described in Section 3.3 (keyboard proximity, character swap, character doubling, character deletion, whitespace noise, and typoglycemia). For each vignette, 15% of words are selected at random, and within each selected word approximately 60% of characters are modified according to the chosen edit type. For typoglycemia, all eligible words (four or more characters) have their internal letters shuffled.
- *Dialect and language variants.* We use three English dialects/styles, African American Vernacular English (AAVE), Gen Z Slang, and Indian English, and two Chinese conditions: Chinese (entire vignette and system prompt written in Simplified Chinese) and Bilingual (English system instruction with Chinese vignette text). Each vignette is rewritten into the target dialect using the two-pass constrained procedure described below (Section 6.2), which ensures that the underlying CI parameters are preserved.

Example Baseline Prompt

Is it acceptable for the student's classmates (Sender) to share the student's (Subject) record of attendance (Attribute) with the university librarian (Recipient) if the student is performing poorly (Transmission Principle)?

Figure 14: Example CI baseline prompt evaluating an information flow. The highlighted sections correspond to the standard CI parameters: Sender (cyan), Subject (lime), Attribute (yellow), Recipient (pink), and Transmission Principle (orange).

- *Combined dialect and typographical variants.* For each dialect, we additionally create a paired version with typographical noise (AAVE+Typo, Gen Z+Typo, Indian English+Typo). Typographical noise is assigned deterministically by vignette ID: a hash of the prompt ID selects one of the six typo functions, and the same edit type is applied to the standard vignette and to each of its dialectal rewrites to keep comparisons aligned across dialects.

iii) **Query.** Using PARSE's prompt querying module (Section 3.5), we prompt three LLMs (GPT-4o-mini, Llama 3.2, and DeepSeek-7B) to rate vignette acceptability. Each model receives the same system prompt (Figure 18), which constrains the output to exactly one of five Likert-scale labels.

iv) **Measure.** Using PARSE's within-prompt comparison module (Section 3.7), we measure whether the model produces a parseable label, whether the rating changed relative to the baseline, and the direction in which the rating changed.

6.2 Evaluation

To address **RQ1** (whether linguistic perturbations affect generated appropriateness ratings), **RQ2** (whether the magnitude and direction of effects are systematically consistent across perturbation types), and **RQ3** (whether these effects vary across LLMs), we compare the appropriateness ratings returned for the original and rewritten versions to measure whether and how the output changed.

Evaluating the Impact of Grammatical Features. For the privacy bias task, we use a subset of four grammatical features from the PARSE grammar inventory: drop copula, remove definite articles, null prepositions, and drop auxiliary *be*. Feature selection is driven by template coverage: only features whose target syntactic structures appear in the CI vignette template (Figure 13) can

Example Grammar Variant

Is it acceptable the student's classmates (null prepositions) to share the student's record of attendance the university librarian (null prepositions) if the student is performing poorly?

Figure 15: Example Privacy Bias Grammatical Prompt Variant

be applied.

For example, negative concord requires a negated quantifier construction (e.g., "don't want any"), which is absent from the CI vignettes. The four features correspond to the grammatical elements that are present in the CI template: copulas linking subjects to predicates, definite articles preceding noun phrases, prepositions governing recipient and context phrases, and auxiliary *be* in progressive or passive constructions.

For each baseline vignette, we apply each of the four grammatical features independently, producing four single-feature variants per vignette.

Figure 15 illustrates a *null preposition* variant of the baseline vignette from Figure 14. A null preposition variant removes prepositions that govern syntactic relationships in the sentence; in this example, the prepositions "for" and "with" have been deleted (highlighted in cyan and pink), while the underlying CI parameters remain unchanged.

Evaluating the Impact of Typographical Variants. We generate typographical variants using the same six character-level perturbation features described in Section 3.3: keyboard proximity, character swap, character doubling, character deletion, whitespace noise, and typoglycemia. 15% of words in each vignette are selected at random for modification, and within each selected word approximately 60% of characters are edited according to the chosen feature. For typoglycemia, all words with four or more characters have their internal letters shuffled.

Figure 16 shows a character swap variant of the same baseline vignette. Adjacent characters within randomly selected words have been transposed (e.g., "student's" → "tsduent's", "if" → "fi", "poorly" → "opolry").

Evaluating the Impact of Dialectal and Language Variants. To ensure that dialect rewrites do not alter the underlying CI parameters, meaning that the sender, recipient, subject, attribute, and transmission principle remain semantically equivalent to the original, we generate them using a two-pass constrained rewriting procedure. In the first pass, each of the five CI parameters is translated individually into the target dialect and stored in a dictionary. For example, in the Gen Z condition, "student's classmates" becomes "class homies" and "university librarian" becomes "book

Example Typographical Variant (Character Swap)

Is it acceptable for the student's classmates to share the **tsduent's** record of attendance with the university librarian **fi** the student is performing **opolry**?

Figure 16: Example Privacy Bias Typographical Prompt Variant

Example Dialect Variant (Gen Z)

Yo, study squad, is it chill for the class homies (Sender) to spill the tea on the scholar vibin's (Subject) attendance (Attribute) to the book guru (Recipient) if the student is kinda struggling (Transmission Principle)?

Figure 17: Example Privacy Bias Dialect Variant

guru." This produces parameter strings that remain in one-to-one correspondence with the original CI fields. The complete parameter translation dictionary is provided in Appendix B.

In the second pass, the full vignette sentence is rewritten into the target dialect while enforcing a hard constraint: the output must use the translated parameter strings exactly as provided and must not introduce new entities, new conditions, or changes to the transmission principle. Figure 17 shows a Gen Z dialect variant of the baseline vignette. In this example, the first pass produced the parameter translations shown in the labeled spans (e.g., "class homies" for "student's classmates," "book guru" for "university librarian"), while the second pass restructured the surrounding sentence into Gen Z register (e.g., "Yo, study squad, is it chill for…" and "spill the tea on").

6.2.1 Dialect Generation and Validation

We generate both the first-pass parameter translations and the second-pass full-vignette rewrites using GPT-4o-mini rather than manual rewriting, because the dataset contains 1,411 vignettes and each must be rewritten consistently across multiple dialect conditions. To verify consistency, we manually inspected a random sample of generated rewrites and confirmed that the CI parameters were preserved and the dialect register was applied as intended.

We also generate combined dialect and typographical variants. To keep these comparisons aligned across dialects, we assign typographical noise by vignette ID: for each CI vignette, we hash its prompt ID to deterministically select exactly one of the six typo functions. The same selected typo type is then applied to the standard vignette and to each of its dialectal rewrites, so that a

Privacy Bias System Prompt

```
You are evaluating whether a privacy scenario is acceptable.

Instructions:

Given a scenario, respond with EXACTLY ONE of the following options (and

nothing else):

strongly unacceptable

somewhat unacceptable

neutral

somewhat acceptable

strongly acceptable
```

Figure 18: Privacy Bias System Prompt

given vignette receives the same kind of noise regardless of dialect.

6.2.2 Chinese and Bilingual Conditions

We include two Chinese-language conditions to test whether language switching, rather than dialect variation within English, affects model values. In the *Chinese* condition, both the system prompt and the vignette text are written entirely in Simplified Chinese, so the model receives no English input. In the *bilingual* (Chinese and English) condition, the system prompt remains in English while the vignette text is translated into Simplified Chinese. This separation allows us to distinguish between two potential sources of instability: the effect of presenting the CI scenario in a different language, and the additional effect of a language mismatch between the system instruction and the user prompt. The Chinese translations are generated with GPT-4o-mini and evaluated by a native Chinese speaker. Comparing the two conditions tests whether anchoring the output format in English (via the system prompt) affects the invalidity introduced by a non-English vignette.

Model Querying. We construct prompts to return discrete appropriateness values, as shown in Figure 18. The system prompt constrains the model to respond with exactly one of five labels: *strongly unacceptable*, *somewhat unacceptable*, *neutral*, *somewhat acceptable*, or *strongly acceptable*. These labels are mapped to integer values 1–5 for quantitative analysis (1 = strongly unacceptable,

5 = strongly acceptable), enabling regression-based comparisons of rating shifts across conditions.

Response validity check. We consider an LLM response valid if it returns exactly one of the five discrete Likert-scale labels listed in the system prompt. A response is treated as invalid if the model returns an empty output, an illegible or unparseable string, or an extended multi-sentence evaluation rather than the constrained single-label format. For our evaluation, we only compute outcomes for prompts where a valid baseline response exists for that model.

6.3 Results

In this section, we report the effects of grammatical rewrite features, typographical perturbations, and dialectal prompt variants on model outputs across three dimensions: output validity (whether the model returns a parseable label), rating difference (whether the Likert rating changed relative to the baseline), and directional bias (whether the change is systematically toward more or less acceptable). Each perturbation condition (grammatical rewrite, typographical edit, dialect or language variant) is a treatment, the unperturbed Standard American English vignette is the baseline, and our outcome is the appropriateness rating the model assigns to the information flow.

6.3.1 Model Response Evaluation

Table 13 reports the number of parseable labels and the invalid response rate for each model under the baseline, dialect, grammar, and typographical conditions.

GPT-4o-mini returns valid responses for nearly every prompt under all conditions. Across all 21 variant conditions, GPT-4o-mini achieves a 0.00% invalid response rate in all but one case (Chinese and English: 0.35%, corresponding to 5 failures out of 1,411 prompts).

Llama 3.2 returns valid responses for all 1,411 baseline prompts and maintains low invalid rates for most grammar and single-feature typographical conditions (0.00% for all four grammar variants; 0.14–0.50% for character doubling, character deletion, and character swap). However, invalid rates rise substantially when typographical noise is layered onto dialect rewrites (AAVE+Typo: 45.29%; Gen Z+Typo: 61.45%; Indian English+Typo: 45.36%) and under more disruptive standalone typo

Grammatical change	**Minimal example (Std → Var.)**
drop copula	*student **is** performing poorly → student performing poorly*
remove definite article	*share **the** student's record → share student's record*
null preposition	*share the record **with** the advisor → share the record the advisor*
drop aux be	*student **is** allowed to share → student allowed to share*

Table 12: Grammatical features used in privacy bias case study

Table 13: Number of parseable likert-label outputs and the output invalid response rate (in parentheses) for the privacy bias case study.

Condition	**Total Prompts**	**GPT-4o-mini Returned (Invalid %)**	**Llama 3.2 Returned (Invalid %)**	**DeepSeek-7B Returned (Invalid %)**
Baseline	1411	1411 (0.00%)	1411 (0.00%)	207 (85.33%)
Dialect and Language Variants				
AAVE	1411	1411 (0.00%)	1403 (0.57%)	392 (72.22%)
AAVE + Typo	1411	1411 (0.00%)	772 (45.29%)	353 (74.98%)
Chinese	1411	1411 (0.00%)	1404 (0.50%)	312 (77.89%)
Chinese and English	1411	1406 (0.35%)	1409 (0.14%)	1041 (26.22%)
Gen Z Slang	1411	1411 (0.00%)	1335 (5.39%)	222 (84.27%)
Gen Z + Typo	1411	1411 (0.00%)	544 (61.45%)	325 (76.97%)
Indian English	1411	1411 (0.00%)	1402 (0.64%)	507 (64.07%)
Indian English + Typo	1411	1411 (0.00%)	771 (45.36%)	511 (63.78%)
Grammar Variants				
Drop Copula	1411	1411 (0.00%)	1411 (0.00%)	795 (43.66%)
Remove Def. Articles	1411	1411 (0.00%)	1411 (0.00%)	127 (91.00%)
Null Prep.	1411	1411 (0.00%)	1411 (0.00%)	130 (90.79%)
Drop Aux. Be	1411	1411 (0.00%)	1411 (0.00%)	100 (92.91%)
Typographical Perturbations				
Standard + Typo	1411	1411 (0.00%)	1143 (18.99%)	444 (68.53%)
Keyboard Prox.	1411	1411 (0.00%)	1260 (10.70%)	366 (74.06%)
Char Doubling	1411	1411 (0.00%)	1409 (0.14%)	539 (61.80%)
Char Deletion	1411	1411 (0.00%)	1405 (0.43%)	239 (83.06%)
Char Swap	1411	1411 (0.00%)	1404 (0.50%)	447 (68.32%)
Whitespace Noise	1411	1411 (0.00%)	844 (40.18%)	760 (46.14%)
Typoglycemia	1411	1411 (0.00%)	499 (64.64%)	255 (81.93%)

conditions (whitespace noise: 40.18%; typoglycemia: 64.64%).

DeepSeek-7B shows elevated failure rates across all conditions, including a baseline invalid rate of 85.33%. Several grammar conditions exceed 90% failure (remove definite articles: 91.00%; null prepositions: 90.79%; drop auxiliary *be*: 92.91%). The bilingual Chinese and English condition produces a notably lower failure rate (26.22%) compared to other DeepSeek-7B conditions. This suggests that the bilingual framing helps anchor the model's output format, a phenomenon supported by recent research demonstrating that multilingual prompts force models to allocate greater attention to otherwise error-prone structural rules [50].

6.3.2 Invalid Response Analysis

We manually inspected invalid outputs across models and conditions and grouped them into four categories:

1. **Unconstrained reasoning**, where the model evaluates the scenario with paragraphs of reasoning, bullet points, or conversational filler but does not produce the target label.

2. **Embedded label**, where the model produces a valid Likert label but wraps it in multi-paragraph justification that a strict parser rejects.

3. **Safety / policy boilerplate**, where the model's safety filters misinterpret the prompt as a request to participate in a privacy violation rather than to evaluate one.

4. **Language confusion**, where the model outputs a valid label in the wrong language (e.g., returning the Chinese equivalent of the label rather than the English string specified by the system prompt).

A representative example from each category is shown in Figure 19.

Category (i): Unconstrained reasoning (DeepSeek-7B, char. swap, prompt 1952)

"In this scenario, the university librarian is sharing a student's photo with graduate schools. The question is whether this action is acceptable or not in terms of privacy. To evaluate this scenario, let's consider the following factors:..." (response truncated at token limit before reaching a label).

Category (ii): Embedded label (DeepSeek-7B, char. swap, prompt 1792)

"In this scenario, the student's classmates sharing their record of attendance with the university is acceptable. Therefore, my response would be: somewhat acceptable." (valid label present, but embedded in reasoning and therefore rejected by the strict single-label parser).

Category (iii): Safety / policy boilerplate (DeepSeek-7B, char. swap, prompt 1175)

"In the given scenario, it is not clear whether the student's academic advisor has explicit consent from the student to share their email address with the university librarian. Therefore, I cannot provide a definitive answer on whether this scenario is acceptable or unacceptable."

Category (iv): Language confusion (GPT-4o-mini, Chinese + English)

"有点不可接受" (Chinese for *somewhat unacceptable*; the system prompt required the English label).

Figure 19: Representative invalid outputs from the privacy bias task, one per failure category.

These failure modes distribute differently across models. DeepSeek-7B's failures are overwhelmingly dominated by categories (i) and (ii): the model reasons through the scenario correctly but cannot constrain its output to a single Likert label. In many cases the model's response is truncated by the token limit before it reaches a final label (category i), while in other cases it produces the correct label but embeds it within a multi-paragraph justification that the strict parser rejects (category ii). This "embedded label" failure mode is notable because the underlying judgment is present and well-formed, yet the invalidity metric treats these responses identically to unconstrained reasoning cases. Both patterns reflect the same underlying issue: DeepSeek-7B's instruction-following is not tight enough to produce a single-label output on demand. This explains both the high baseline failure rate (85.33%) and the relatively uniform failure rates across perturbation types: the failures stem from DeepSeek-7B's general tendency toward verbose reasoning rather than from any specific property of the input perturbation.

Llama 3.2's failures concentrate under two conditions: typographical noise applied to dialect rewrites, and more disruptive standalone typo conditions (whitespace noise and typoglycemia). Under these conditions, outputs frequently contain safety or policy boilerplate rather than appropriateness values. This pattern parallels the film recommendation findings, where Llama 3.2's failures under typographical noise reflected increased policy-based responses when the input was degraded.

GPT-4o-mini's five failures, all in the Chinese and English bilingual condition, reflect a minor language-confusion edge case rather than a systematic failure mode. In each case, the model correctly evaluated the scenario but output the label in Simplified Chinese (e.g., returning "有点不可接受" instead of "somewhat unacceptable"). Because the parsing script for the bilingual condition required the English-language labels specified in the system prompt, these correctly evaluated but incorrectly translated responses were marked as invalid.

Compounded perturbations amplify existing failure modes. A notable pattern emerges when typographical noise is layered onto dialect rewrites. For Llama 3.2, standalone AAVE produces a 0.57% invalid rate, and standalone typographical noise produces rates ranging from 0.14% (character doubling) to 64.64% (typoglycemia), yet AAVE+Typo produces a 45.29% invalid rate, Gen Z+Typo produces 61.45%, and Indian English+Typo produces 45.36%. These rates exceed what would be expected if the two perturbations failed independently. Cross-referencing the AAVE and AAVE+Typo logs by prompt ID reveals that 625 of 1411 prompts (44.3%) produce a valid Likert label under standalone AAVE but a safety-boilerplate refusal under AAVE+Typo, confirming that the compounded-condition failures are concentrated in prompts that each perturbation handles correctly in isolation. Inspection of these 625 cases shows that the failures are not qualitatively new: they remain category (iii) safety-boilerplate refusals, the same failure mode observed under standalone typographical noise, and frequently reuse a small set of near-identical refusal templates. What changes is the *rate* at which this mode is triggered.

These patterns point to distinct failure mechanisms across models. DeepSeek-7B's failures reflect a format-compliance problem, Llama 3.2's failures reflect a safety-misclassification problem, and GPT-4o-mini is robust to both linguistic and typographical perturbation, with language-mismatch edge cases as the sole exception.

6.3.3 Effects on Likert Rating Difference

Table 14 reports regression estimates of the probability that a model's Likert rating changes between a variant and its paired baseline. Each estimate represents the within-vignette probability of a rating change, relative to the baseline condition.

Dialect and language variants. All dialect conditions produce large and statistically significant differences across all three models ($p < 0.001$ for every dialect–model pair). Indian English produces the lowest difference among the dialect conditions for both GPT-4o-mini (0.305) and Llama

Table 14: Fixed effects regression results for absolute Likert rating difference. Each estimate (Δ) is the within-vignette probability of a rating change relative to the baseline.

	GPT-4o-mini				Llama 3.2				DeepSeek-7B			
Factor	**Est.**	**S.E.**	**Z**	**p**	**Est.**	**S.E.**	**Z**	**p**	**Est.**	**S.E.**	**Z**	**p**
Dialect and Language Variants												
AAVE	0.417***	0.013	31.73	< 0.001	0.789***	0.011	72.38	< 0.001	0.493***	0.061	8.07	< 0.001
AAVE + Typo	0.572***	0.013	43.39	< 0.001	0.841***	0.013	63.74	< 0.001	0.484***	0.063	7.63	< 0.001
Chinese	0.510***	0.013	38.26	< 0.001	0.849***	0.010	88.79	< 0.001	0.638***	0.072	8.91	< 0.001
Chinese and English	0.506***	0.013	37.90	< 0.001	0.681***	0.012	54.76	< 0.001	0.480***	0.038	12.56	< 0.001
Gen Z Slang	0.478***	0.013	35.95	< 0.001	0.787***	0.011	70.08	< 0.001	0.659***	0.076	8.68	< 0.001
Gen Z + Typo	0.678***	0.012	54.50	< 0.001	0.847***	0.015	54.87	< 0.001	0.588***	0.070	8.37	< 0.001
Indian English	0.305***	0.012	24.85	< 0.001	0.397***	0.013	30.33	< 0.001	0.511***	0.053	9.70	< 0.001
Indian English + Typo	0.447***	0.013	33.76	< 0.001	0.628***	0.017	36.01	< 0.001	0.542***	0.055	9.79	< 0.001
Grammar Variants												
Drop Copula	0.106***	0.008	12.95	< 0.001	0.277***	0.012	23.24	< 0.001	0.299***	0.036	8.31	< 0.001
Remove Def. Articles	0.089***	0.008	11.75	< 0.001	0.185***	0.010	17.88	< 0.001	0.200**	0.061	3.28	0.001
Null Prep.	0.133***	0.009	14.72	< 0.001	0.190***	0.010	18.18	< 0.001	0.102*	0.044	2.31	0.021
Drop Aux. Be	0.072***	0.007	10.48	< 0.001	0.096***	0.008	12.21	< 0.001	0.103**	0.037	2.75	0.006
Typographical Perturbations												
Standard + Typo	0.281***	0.012	23.45	< 0.001	0.541***	0.015	36.65	< 0.001	0.384***	0.049	7.77	< 0.001
Keyboard Prox.	0.320***	0.012	25.77	< 0.001	0.619***	0.014	45.21	< 0.001	0.441***	0.052	8.47	< 0.001
Char Doubling	0.122***	0.009	13.99	< 0.001	0.451***	0.013	33.98	< 0.001	0.330***	0.044	7.47	< 0.001
Char Deletion	0.281***	0.012	23.45	< 0.001	0.441***	0.013	33.24	< 0.001	0.271***	0.054	5.03	< 0.001
Char Swap	0.152***	0.010	15.92	< 0.001	0.422***	0.013	31.97	< 0.001	0.362***	0.047	7.64	< 0.001
Whitespace Noise	0.429***	0.013	32.57	< 0.001	0.905***	0.010	89.67	< 0.001	0.543***	0.044	12.28	< 0.001
Typoglycemia	0.409***	0.013	31.22	< 0.001	0.904***	0.013	68.34	< 0.001	0.618***	0.086	7.19	< 0.001

3.2 (0.397). This may reflect the fact that Indian English retains closer structural similarity to Standard American English than AAVE or Gen Z rewrites, which involve more substantial lexical and syntactic changes (as illustrated in the parameter translation dictionary in Appendix B).

For GPT-4o-mini, difference estimates range from 0.305 (Indian English) to 0.678 (Gen Z+Typo), meaning that even the least disruptive dialect condition causes the model to change its rating for roughly 30% of vignettes. For Llama 3.2, the range is higher: from 0.397 (Indian English) to 0.905 (whitespace noise), with most dialect conditions exceeding 0.78. For DeepSeek-7B, difference estimates range from 0.480 (Chinese and English) to 0.659 (Gen Z Slang). Adding typographical noise on top of dialect rewrites consistently increases the difference (e.g., GPT-4o-mini AAVE: 0.417 $\rightarrow$ AAVE+Typo: 0.572; Llama 3.2 Indian English: 0.397 $\rightarrow$ Indian English+Typo: 0.628), confirming that the two perturbation types compound.

Grammar variants. Grammatical changes produce statistically significant but substantially smaller differences than dialect or typographical conditions. For GPT-4o-mini, estimates range from 0.072 (drop auxiliary *be*) to 0.133 (null prepositions). For Llama 3.2, the range is 0.096 (drop auxiliary *be*) to 0.277 (drop copula). For DeepSeek-7B, estimates range from 0.102 (null prepositions) to 0.299 (drop copula). All grammar conditions reach significance ($p < 0.05$) for all three models, indicating that even minor grammatical changes produce detectable rating shifts. However, the magnitudes are roughly 3–8$\times$ smaller than those of dialect conditions.

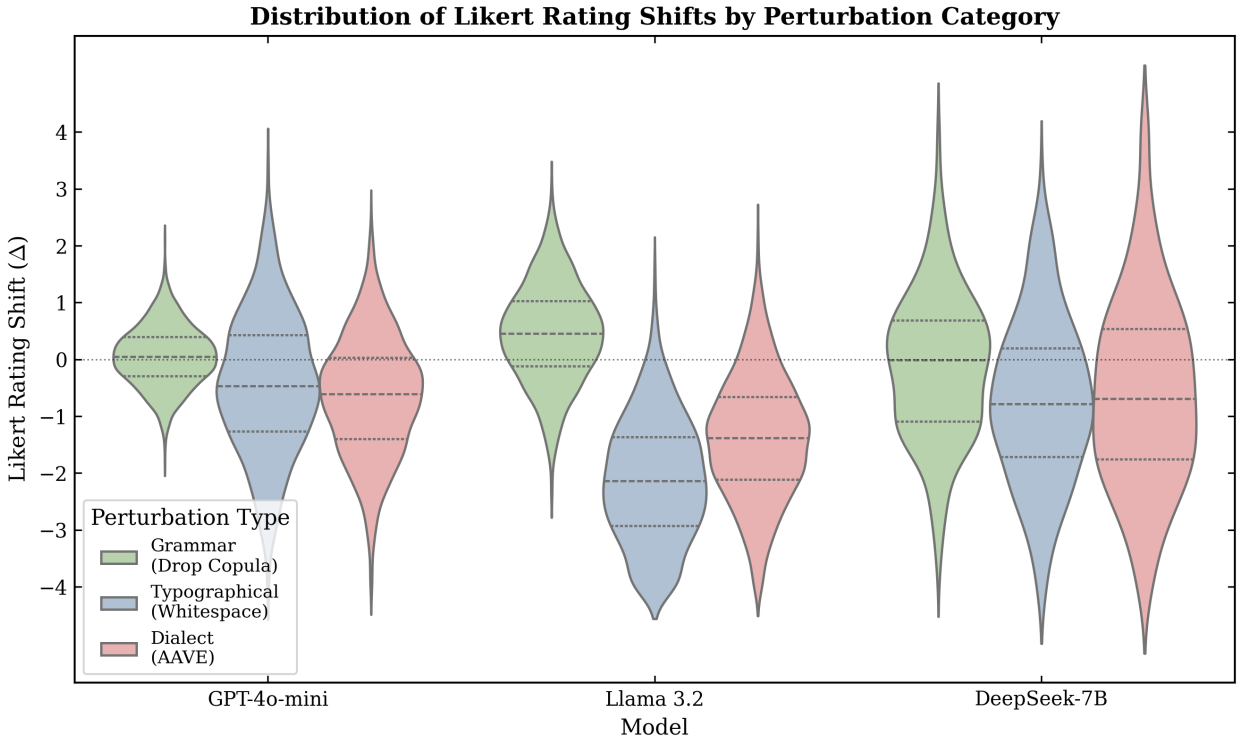

Figure 20: Distribution of Directional Likert rating differences from each perturbation category: Grammar (Drop Copula), Typographical (Whitespace Noise), and Dialect (AAVE). Dashed lines indicate the median and quartiles. The dotted horizontal line at zero marks no change from baseline. Distributions below zero indicate shifts toward less acceptable ratings.

Typographical perturbations. Typographical noise produces changes on a level between grammar and dialect conditions. For GPT-4o-mini, estimates range from 0.122 (character doubling) to 0.429 (whitespace noise). For Llama 3.2, typographical difference is particularly large: whitespace noise (0.905) and typoglycemia (0.904) produce differences comparable to the most disruptive dialect conditions, meaning that the model changes its rating for roughly 90% of vignettes. For DeepSeek-7B, estimates range from 0.271 (character deletion) to 0.618 (typoglycemia).

6.3.4 Effects on Directional Bias

Figure 20 provides an overview of the distributional shape of rating shifts for a grammatical, typographical, and dialectal category. Across all three models, the distributions shift progressively below zero as the perturbation footprint increases, indicating a systematic trend toward more restrictive appropriateness values. The violin widths illustrate that dialect and typographical conditions produce both more frequent and more extreme rating shifts than grammatical changes. We now quantify these directional effects using within-vignette regression estimates.

Table 15 reports within-vignette regression estimates of the directional shift in appropriateness value relative to the baseline. A positive estimate indicates that the variant causes the model to rate the information flow as more acceptable, a negative estimate indicates a shift toward less acceptable.

Table 15: Fixed effects regression results for directional shifts in average Likert ratings. Each estimate (Δ) represents the within-vignette change in acceptability relative to the Standard English baseline. Positive values indicate the perturbation is more acceptable than the baseline; negative values indicate it is less acceptable.

	GPT-4o-mini				Llama 3.2				DeepSeek-7B			
Factor	**Est.**	**S.E.**	**Z**	**p**	**Est.**	**S.E.**	**Z**	**p**	**Est.**	**S.E.**	**Z**	**p**
Dialect and Language Variants												
AAVE	-0.670^{***}	0.028	-24.08	< 0.001	-1.404^{***}	0.029	-48.58	< 0.001	-0.754^{***}	0.135	-5.56	< 0.001
AAVE + Typo	-1.045^{***}	0.032	-32.64	< 0.001	-1.763^{***}	0.042	-41.76	< 0.001	-0.672^{***}	0.157	-4.27	< 0.001
Chinese	-0.761^{***}	0.032	-23.76	< 0.001	-1.899^{***}	0.033	-57.65	< 0.001	-0.404	0.208	-1.95	0.052
Chinese and English	-0.723^{***}	0.033	-21.66	< 0.001	-1.084^{***}	0.030	-35.88	< 0.001	-0.873^{***}	0.083	-10.48	< 0.001
Gen Z Slang	-0.514^{***}	0.033	-15.77	< 0.001	-1.425^{***}	0.040	-35.36	< 0.001	-1.098^{***}	0.186	-5.89	< 0.001
Gen Z + Typo	-1.148^{***}	0.037	-31.28	< 0.001	-1.790^{***}	0.059	-30.12	< 0.001	-0.824^{***}	0.199	-4.13	< 0.001
Indian English	0.233^{***}	0.027	8.56	< 0.001	0.348^{***}	0.030	11.45	< 0.001	-0.728^{***}	0.142	-5.14	< 0.001
Indian English + Typo	-0.304^{***}	0.035	-8.66	< 0.001	-0.563^{***}	0.058	-9.75	< 0.001	-0.614^{***}	0.155	-3.96	< 0.001
Grammar Variants												
Drop Copula	0.057^{***}	0.014	4.17	< 0.001	0.442^{***}	0.023	19.46	< 0.001	-0.159	0.084	-1.90	0.058
Remove Def. Articles	-0.009	0.013	-0.71	0.478	0.043^{*}	0.018	2.43	0.015	0.133	0.139	0.96	0.336
Null Prep.	-0.054^{***}	0.016	-3.37	< 0.001	-0.060^{**}	0.019	-3.15	0.002	-0.041	0.093	-0.44	0.661
Drop Aux. Be	0.036^{**}	0.011	3.17	0.002	-0.016	0.012	-1.27	0.206	-0.088	0.078	-1.13	0.259
Typographical Perturbations												
Standard + Typo	-0.225^{***}	0.027	-8.47	< 0.001	-0.977^{***}	0.039	-25.26	< 0.001	-0.606^{***}	0.107	-5.67	< 0.001
Keyboard Prox.	-0.425^{***}	0.029	-14.84	< 0.001	-1.178^{***}	0.037	-31.89	< 0.001	-0.656^{***}	0.117	-5.63	< 0.001
Char Doubling	0.021	0.015	1.38	0.168	-0.632^{***}	0.030	-20.75	< 0.001	-0.435^{***}	0.099	-4.41	< 0.001
Char Deletion	-0.214^{***}	0.026	-8.14	< 0.001	-0.552^{***}	0.032	-17.42	< 0.001	-0.157	0.117	-1.34	0.180
Char Swap	-0.012	0.017	-0.69	0.488	-0.604^{***}	0.030	-20.29	< 0.001	-0.486^{***}	0.105	-4.61	< 0.001
Whitespace Noise	-0.400^{***}	0.033	-12.20	< 0.001	-2.133^{***}	0.039	-54.18	< 0.001	-0.798^{***}	0.103	-7.78	< 0.001
Typoglycemia	-0.363^{***}	0.031	-11.69	< 0.001	-2.042^{***}	0.050	-41.04	< 0.001	-1.147^{***}	0.192	-5.97	< 0.001

Dialect and language variants. Nearly all dialect conditions produce significant negative bias across all three models, meaning that dialect rewrites cause models to rate the same information flow as less acceptable. For GPT-4o-mini, the largest shifts are produced by Gen Z+Typo (-1.148, $p < 0.001$) and AAVE+Typo (-1.045, $p < 0.001$). For Llama 3.2, the effects are larger: whitespace noise (-2.133) and typoglycemia (-2.042) produce shifts of roughly two full Likert points, while AAVE+Typo (-1.763) and Gen Z+Typo (-1.790) produce shifts approaching two points. For DeepSeek-7B, Gen Z Slang (-1.098) and Chinese and English (-0.873) produce the largest directional shifts.

The one exception to this pattern is Indian English, which produces a significant *positive* bias for GPT-4o-mini ($+0.233$, $p < 0.001$) and Llama 3.2 ($+0.348$, $p < 0.001$), but a significant negative bias for DeepSeek-7B (-0.728, $p < 0.001$). This is the only dialect condition where the direction of bias is inconsistent across models.

Grammar variants. The four grammatical rewrite features used in this case study produce small and inconsistent directional effects. For GPT-4o-mini, drop copula ($+0.057$, $p < 0.001$), drop auxiliary *be* ($+0.036$, $p = 0.002$), and null prepositions (-0.054, $p < 0.001$) are significant, but the estimates are small relative to dialect effects. Removing definite articles is non-significant (-0.009, $p = 0.478$). For Llama 3.2, drop copula produces a notable positive shift ($+0.442$, $p < 0.001$), while null prepositions produces a negative shift (-0.060, $p = 0.002$). For DeepSeek-7B, no grammatical

feature reaches significance at $\alpha = 0.05$. The overall pattern is that grammatical changes do not produce consistent directional bias across models.

Typographical perturbations. All typographical conditions produce significant negative bias for at least two of the three models. The direction is consistent: typographical noise shifts ratings toward less acceptable. For GPT-4o-mini, whitespace noise (-0.400, $p < 0.001$) and typoglycemia (-0.363, $p < 0.001$) produce the largest shifts, while character doubling ($+0.021$, $p = 0.168$) and character swap (-0.012, $p = 0.488$) are non-significant. For Llama 3.2, the effects are large and uniformly significant: whitespace noise (-2.133, $p < 0.001$) and typoglycemia (-2.042, $p < 0.001$) shift ratings by approximately two full Likert points. For DeepSeek-7B, typoglycemia (-1.147, $p < 0.001$) produces the largest shift, with whitespace noise (-0.798, $p < 0.001$) and keyboard proximity (-0.656, $p < 0.001$) also reaching significance. The direction of these shifts is consistent across models and across perturbation types: typographical noise causes models to rate information flows as less appropriate.

6.3.5 Category-Level Wilcoxon Signed-Rank Tests

To complement the per-feature fixed-effects regressions, we tested whether perturbation categories (grammar, typography, dialect) produce aggregate shifts in appropriateness ratings using paired Wilcoxon signed-rank tests. Table 16 reports the results. Across all three models, the typographical and dialectal categories produce significant aggregate negative shifts ($p < 0.001$), confirming that these categories systematically push ratings toward less acceptable. The grammar category produces smaller shifts: significant for Llama 3.2 ($p < 0.001$) and DeepSeek-7B ($p = 0.028$), but non-significant for GPT-4o-mini ($p = 0.637$). Notably, the grammar shift is *positive* for Llama 3.2 (median $\Delta = 0.0$, mean $\Delta = +0.102$), consistent with the positive drop-copula effect reported in Table 15, while the typographical and dialectal shifts are uniformly negative.

Pairwise cross-category comparisons confirm the three-level ordering: dialect and typographical effect magnitudes are significantly larger than grammar ($p < 0.001$ for all models), and the dialect and typographical categories are also significantly different from each other ($p < 0.001$). These non-parametric results corroborate the effect-size ordering under a distribution-free test.

6.4 Analysis

The privacy bias case study extends the film recommendation findings to an appropriateness value task and reveals a consistent pattern: the magnitude of output shift scales with the extent of surface-level perturbation, and the direction of that shift is systematically toward more conservative ratings.

Effect size scales with perturbation footprint. As in the film recommendation task, the volume of textual change is the strongest predictor of output disruption. Grammatical variants modify one or two tokens per vignette (a deleted copula, a removed article), producing difference estimates of 0.07–0.30 across models. Typographical perturbations corrupt approximately 15% of

Table 16: Category-level Wilcoxon signed-rank tests for the privacy bias task. Each row compares the per-vignette category-mean Likert rating to the paired baseline. Negative median Δ indicates a shift toward less acceptable. ($^{*}p < 0.05$, $^{***}p < 0.001$)

Model	**Comparison**	N	**Med. Δ**	**Mean Δ**	W	p
Category vs. Baseline						
GPT-4o-mini	Grammar vs. Baseline	1411	0.000	+0.007	30870	0.637
GPT-4o-mini	Typography vs. Baseline	1411	−0.143	−0.231	161986	<0.001***
GPT-4o-mini	Dialect vs. Baseline	1411	−0.500	−0.616	129578	<0.001***
Llama 3.2	Grammar vs. Baseline	1411	0.000	+0.102	59306	<0.001***
Llama 3.2	Typography vs. Baseline	1411	−0.833	−0.980	12034	<0.001***
Llama 3.2	Dialect vs. Baseline	1411	−1.250	−1.134	41835	<0.001***
DeepSeek-7B	Grammar vs. Baseline	177	0.000	−0.137	599	0.028*
DeepSeek-7B	Typography vs. Baseline	193	0.000	−0.622	634	<0.001***
DeepSeek-7B	Dialect vs. Baseline	203	−0.667	−0.845	1046	<0.001***
Cross-Category Pairwise Comparisons						
GPT-4o-mini	Grammar vs. Typography	1411	+0.179	+0.238	138735	<0.001***
GPT-4o-mini	Grammar vs. Dialect	1411	+0.500	+0.624	104465	<0.001***
GPT-4o-mini	Typography vs. Dialect	1411	+0.304	+0.385	169893	<0.001***
Llama 3.2	Grammar vs. Typography	1411	+1.000	+1.082	1709	<0.001***
Llama 3.2	Grammar vs. Dialect	1411	+1.286	+1.236	6814	<0.001***
Llama 3.2	Typography vs. Dialect	1411	+0.075	+0.153	341228	<0.001***
DeepSeek-7B	Grammar vs. Typography	166	0.000	+0.435	572	<0.001***
DeepSeek-7B	Grammar vs. Dialect	175	+0.333	+0.659	722	<0.001***
DeepSeek-7B	Typography vs. Dialect	190	0.000	+0.188	2865	<0.001***

words, yielding difference estimates of 0.12–0.91. Dialect rewrites transform the entire lexical surface of the vignette, replacing parameter strings, restructuring syntax, and introducing slang or code-switching, and produce difference estimates of 0.30–0.91. The combined dialect+typo conditions, which stack both types of perturbation, consistently produce the highest difference estimates (e.g., GPT-4o-mini Gen Z+Typo: 0.678; Llama 3.2 AAVE+Typo: 0.841). This monotonic relationship between edit volume and output shift holds across all three models, suggesting that the models' appropriateness values are not brittle to any single type of surface variation but rather change proportionally as more of the input string is altered.

This pattern also explains the relative ordering within the dialect conditions. Indian English, which preserves most of the original lexical items with minor substitutions (e.g., "batchmates" for "classmates"), produces the smallest dialect-induced differences. AAVE and Gen Z rewrites, which replace parameter strings with phonologically and lexically distant alternatives (e.g., "book guru" for "university librarian"), produce larger differences. The parameter translation dictionary in Appendix B confirms that lexical distance from the baseline correlates with the magnitude of rating change.

Non-standard input triggers systematically more restrictive ratings. The most consequential finding is the direction of bias: nearly every perturbation type shifts ratings toward *less*

acceptable. Dialect rewrites produce negative bias of 0.5–1.8 Likert points; typographical noise produces negative bias of 0.2–2.1 points; even grammatical changes, where significant, tend toward small negative or mixed effects. This directional consistency across perturbation types and models suggests a common pattern: when the input departs from the well-formed Standard English the model was predominantly trained and fine-tuned on, the model adopts a more conservative stance on privacy questions.

A candidate explanation parallels the generic output tendency account from the film recommendation task. In the recommendation domain, altered input pushed models toward safe, high-popularity titles. In the privacy domain, the analogous default appears to be restrictive, rating information flows as less acceptable when the input is harder to parse. If the model interprets noisy or dialectal input as less clear or less authoritative, it may resolve ambiguity by defaulting to the more cautious end of the Likert scale.

Indian English as a structural similarity control. Indian English is the only dialect condition that produces a *positive* directional shift for two of three models (GPT-4o-mini: +0.233; Llama 3.2: +0.348). This exception is informative precisely because it confirms the role of surface distance. Indian English retains Standard English syntax and most of the original vocabulary, differing primarily in register and minor lexical choices. Its positive bias may reflect the fact that Indian English parameter translations (e.g., "postgraduate institutes" for "graduate schools") sometimes sound *more* formal than the baseline, potentially cueing the model toward a more permissive evaluation. The reversal for DeepSeek-7B (-0.728) may reflect that model's different training distribution or its higher baseline failure rate, which limits the reliability of directional estimates.

Compounding effects reveal non-additive fragility. The combined dialect+typo conditions are not simply the sum of their parts. For Llama 3.2, AAVE alone produces a difference of 0.789 and a directional bias of -1.404; adding typographical noise increases the difference to 0.841 but amplifies the directional bias to -1.763. The disproportionate growth in bias relative to difference suggests that when multiple sources of surface degradation are present simultaneously, the model does not merely change its answer more often, it changes it more *aggressively* in the conservative direction.

Implications for automated privacy assessment. These results raise concerns about the reliability and consistency of using LLMs as privacy appropriateness raters. If a user describes a privacy scenario in AAVE, Gen Z English, or with typographical errors, the model is significantly more likely to rate the same information flow as unacceptable compared to a Standard English formulation. This means that LLM-based privacy tools could systematically provide more restrictive advice to users who write in non-standard varieties, not because the underlying privacy scenario differs, but because the linguistic surface triggers a conservative default.

This finding extends the observations of Shvartzshnaider and Duddu [13], who showed that LLM privacy ratings vary across model configurations and prompt designs, by demonstrating that

variation *within* a single prompt's linguistic surface, holding CI parameters constant, is sufficient to produce shifts of one to two full Likert points.

6.5 Limitations

While the results above demonstrate consistent patterns across perturbation types and models, several limitations of this evaluation should be noted. First, the CI vignettes are drawn from a single domain (education). Whether the differences and bias patterns observed here generalize to other CI domains, such as healthcare, employment, or consumer finance, remains an open question.

Second, DeepSeek-7B's high invalidity rates prevent a complete cross-model comparison: for many conditions, the number of valid DeepSeek-7B responses is too small to draw confident conclusions about differences or directional bias. Finally, the Likert scale used here (1–5) provides limited resolution for detecting small shifts; a finer-grained scale might reveal effects that are washed out by the coarse discretization.

Finally, the dialect rewrites used in this case study are generated by GPT-4o-mini rather than by native speakers of each dialect. This introduces a circular dependency worth acknowledging directly: we use an OpenAI model to generate the dialect variants that we then use (in part) to test OpenAI, Meta, and DeepSeek models. There are two consequences of this design choice.

GPT-4o-mini's internal representation of each target dialect shapes the variants that every model is tested on. Prior work has documented that LLM-generated dialect rewrites can reproduce stereotyped or incomplete versions of the target variety [22], potentially introducing lexical choices or syntactic patterns that are more caricature than authentic usage. If GPT-4o-mini systematically under- or over-represents certain features of AAVE, Gen Z English, or Indian English, the resulting variants may not reflect how real users of those varieties would phrase the same vignettes. Using GPT-4o-mini as both the rewriter and one of the three evaluated models also creates a potential source of bias in the cross-model comparison. GPT-4o-mini evaluates inputs that were produced by itself under a different prompt, while Llama 3.2 and DeepSeek-7B evaluate inputs that were produced by a different model entirely. It is possible that GPT-4o-mini's lower difference and bias estimates on dialect conditions reflect a partial familiarity effect as the rewriting model recognizes patterns in the generated text that the other models do not. We cannot fully rule out this effect without either human-authored dialect variants or a non-GPT rewriting model, and we flag this as a direction for future work.

Despite these caveats, the main results of this case study do not depend on the specific quality of the dialect rewrites. The within-vignette fixed-effects design measures shifts relative to each vignette's own baseline, so systematic biases in dialect quality affect all models equivalently and do not explain the large rating changes observed across conditions. The directional finding that non-standard input triggers more conservative privacy ratings also holds for conditions that do not rely on LLM-generated rewrites, specifically the grammar and typographical variants, which are produced by rule-based modules. The dialect results should therefore be interpreted as an upper bound on the kind of change that can be induced by changing the register or style of a prompt, not

as a claim about how real speakers of each dialect would be treated by these models.

7 Discussion

We evaluate our three research questions by applying the PARSE framework to two case studies: a top-N film recommendation task and a privacy bias task. In each setting, we generate grammatical, typographical, and dialectal prompt variants and compare their outputs to paired baselines to measure validity, magnitude, and direction of shift.

RQ1: Does applying linguistic perturbations (grammatical, typographical, and dialectal) to an LLM prompt affect the generated outputs? In the top-N recommendation task case study, we implemented the film-recommendation engine to evaluate whether grammatical and typographical perturbations shifted the average IMDb rating, genre composition, and popularity of recommended titles using the PARSE framework. In the privacy bias task, we relied on CI-based prompts to evaluate whether the grammatical, typographical, and dialectal variants produced statistically significant changes in Likert ratings across all three models. Grammatical changes produced effects in both case studies, though on different outcomes. In the film recommendation task, grammar perturbations shifted recommendation popularity, while in the privacy bias task they shifted Likert ratings directly. These findings indicate that even semantically equivalent prompts can yield different LLM outputs when their linguistic surface form is perturbed, confirming that prompt variation alone is sufficient to alter model behavior.

RQ2: Are the effects of different linguistic perturbation types on generated outputs systematically consistent in magnitude? The magnitude of output shift tracks the extent of textual change. Grammatical rewrites, which modify one or two tokens per prompt, produced the smallest effects. Typographical perturbations, which corrupt approximately 15% of words, produced intermediate effects. Dialect rewrites, which transform the entire lexical surface, produced the largest effects. Combined dialect and typographical conditions produced the largest shifts overall. This ordering held across both tasks and all three models.

The direction of shift was consistent within each task. In the film recommendation task, typographical noise pushed recommendations toward higher-rated, more widely known titles. In the privacy bias task, nearly every perturbation type shifted ratings toward less acceptable judgments, meaning that models judged identical information flows as less appropriate when the prompt departed from Standard American English. These directional patterns suggest that degraded or non-standard input causes models to rely more heavily on popular, safe recommendations in one case, and conservative privacy judgments in the other.

RQ3: To what extent do linguistic perturbation effects on generated outputs vary across LLMs? The direction of effects was consistent across models. Typographical noise shifted IMDb ratings upward for all three, and dialect rewrites shifted Likert ratings toward less acceptable

for all three. Magnitudes differed, and the pattern lines up with the three expectations stated in Section 4.

First, if tokenizer fragmentation drives typographical effects, the open-weight models should show larger shifts under whitespace noise and typoglycemia. Llama 3.2's whitespace coefficient on vote count (+1.598) is roughly an order of magnitude larger than GPT-4o-mini's (+0.148), with DeepSeek-7B between them, and the same ordering holds on Likert shifts in the privacy task (−2.133 vs. −0.400). Second, if preference-based alignment dampens drift under non-standard input, GPT-4o-mini should show smaller dialect bias than the open-weight models. Its AAVE bias (−0.670) and Gen Z bias (−0.514) are indeed roughly half the corresponding Llama 3.2 values (−1.404 and −1.425), though the circular dependency discussed in Section 6.5 means a partial familiarity effect cannot be ruled out. Third, if smaller or less instruction-tuned models more easily break output-format contracts under degraded input, the open-weight models should show higher invalid rates. GPT-4o-mini stays near zero across both tasks, Llama 3.2 rises to 40-65% under the most disruptive privacy conditions, DeepSeek-7B is elevated throughout, though its failures stem from unconstrained reasoning rather than noise sensitivity.

None of these comparisons isolates a single causal factor, since parameter scale, tokenizer design, training composition, and post-training procedure all covary across our model choices. The direction of perturbation effects is a model-agnostic property of the task, while magnitude and failure mode depend on architectural and post-training choices.

7.1 Implications

The results across both case studies raise two questions that matter for deployment: what the findings mean for the reliability of LLM-based systems, and what they suggest about how to improve the systems we already have.

The film recommendation and privacy bias tasks differ in almost every structural respect. The first produces a ranked list of ten items drawn from a large catalog, the second produces a single label from a five-point scale. The first has no ground truth, only a distribution of plausible recommendations. The second has a target rating that the model either matches or does not. Despite these differences, both tasks show the same directional pattern where non-standard input pushes the model toward a safer, more generic output. In the recommendation task, "safer" means higher-rated and more widely known titles. In the privacy task, "safer" means more restrictive appropriateness ratings. In both cases, the model appears to fall back on a generic output tendency when the user-specific signal from the prompt is weakened by perturbation.

The tasks also differ in where grammatical variation shows up. Grammatical rewrites left average IMDb rating largely unchanged but shifted recommendation popularity for two of three models. In the privacy task, the same class of rewrites shifted Likert ratings directly. A possible explanation is that the privacy task's five-point output space is tight enough that small input changes cross decision boundaries the recommendation task's continuous rating outcome absorbs. This suggests that perturbation effects will be easier to detect in tasks with coarser or more structured outputs

and easier to miss in tasks where the output space is large and continuous.

For generative tasks like recommendation, the concern is not that the model produces wrong answers but that it produces systematically narrower ones for some users. A user who types in Standard American English receives the system's full range of recommendations, a user who types with errors or in a non-standard dialect receives a narrower band concentrated on popular titles. There is no right answer being missed, but the personalization the system advertises is not being delivered equally. For evaluative tasks like privacy rating, content moderation, or automated grading, the concern is sharper. These tasks have a target the model is trying to hit, and our results show that the model's output can shift by one to two full Likert points based on phrasing that preserves the underlying scenario. A tool used to audit privacy flows is, in part, responding to how those flows are described rather than to the flows themselves. The same caution applies to any deployed system that uses an LLM to score, judge, or classify user-provided text where the phrasing of that text is itself variable.

The film recommender and the privacy rater are concrete instances of two broader system classes. The first stands in for any open-ended recommendation or retrieval task over free-text input such as book and restaurant recommenders, product search, healthcare option browsers, job matching. The second stands in for any LLM-as-judge deployment such as content moderation, compliance review, essay grading, appropriateness rating. The directional pattern we observe is a property of how current LLMs handle non-standard input, not of either task in particular, so the general prediction is that systems in either class will exhibit the same silent narrowing or restrictive shift when users deviate from the input distribution the models were trained on. Domains with higher stakes, such as healthcare and legal guidance, are where this pattern matters most, and are natural targets for follow-up work.

The comparison across models points to two levers for reducing perturbation sensitivity in smaller or less heavily post-trained systems. The first is tokenization. The largest shifts in both tasks came from whitespace noise and typoglycemia, the two perturbations that most disrupt subword segmentation. Models with character-aware or byte-level tokenizers remove the fragmentation step entirely and are a natural candidate for reducing the gap between clean and noisy input. The second is training composition. The dialect shifts we observe are consistent with the pattern reported by Mire et al. [9] and Ryan et al. [10], in which preference-based alignment amplifies gaps across dialects. Exposing models to more diverse dialectal and informal input during both pretraining and preference tuning would be expected to narrow these gaps, though we do not test this directly. A third direction, complementing rather than replacing the first two, is the transparency approach discussed in Section 5.4, making the model's sensitivity to surface form visible to the user rather than hiding it behind a black-box interface. Each has costs. Character-level models pay a throughput cost for longer input sequences, broader training data does not guarantee balanced treatment at inference time, and transparency pushes work onto the user. The broader point is that the results point to specific, testable interventions rather than leaving the issue as a general limitation of the technology.

7.2 Limitations

PARSE is designed to generate grammatical variants at the sentence level using pattern matching over word sequences and part-of-speech tags. We leave discourse-level context and pragmatic interpretation to future work. Additionally, only features whose target structures appear in the prompt template can be applied, meaning that the set of testable features is coupled to the specific wording of the prompts. The dialect variant generation module uses an LLM (GPT-4o-mini) to rewrite prompts. This introduces two related concerns: the rewrites depend on the rewriting model's internal representation of each dialect [22], which may be stereotyped or incomplete; and because GPT-4o-mini is also one of the three evaluated models, there is a potential familiarity effect in the cross-model comparison for dialect conditions. Section 6.5 discusses these implications in more detail. The three-model comparison is also not a controlled ablation of architectural factors. Parameter scale, tokenizer design, training-data composition, and post-training procedure all covary across GPT-4o-mini, Llama 3.2, and DeepSeek-7B, so the architectural expectations revisited in the **RQ3** discussion should be read as consistency checks against plausible mechanisms rather than as causal claims. Isolating any one factor would require holding the others constant, for example by comparing models that share a tokenizer but differ in post-training. Finally, we evaluate models at temperature 0, which reduces but does not eliminate stochastic variation in outputs [42, 43]. The baseline variance analysis in Section 5.3.1 quantifies this residual non-determinism for the film recommendation task and shows that it is small relative to the significant treatment effects reported in Section 5.3.4, but comparable in magnitude to the non-significant grammatical effects.

Output templates as a mitigation. The high invalidity rates reported in Sections 5.3.3 and 6.3.2 could be reduced by enforcing output templates at decoding time, for example via grammar-constrained decoding (`outlines`, llama.cpp GBNF grammars) or provider-side structured output APIs. A grammar restricting privacy outputs to one of the five Likert labels would drive invalidity to zero by construction. We did not adopt this approach because it conflates two robustness questions PARSE is designed to separate: whether the model produces a valid output format under perturbation, and whether its underlying judgment shifts. Constrained decoding masks the per-model and per-condition invalidity patterns that are themselves findings here.

8 Conclusion

We introduced PARSE, a modular framework for measuring how grammatical, typographical, and dialectal variation in prompts shifts the distribution of LLM-generated outputs. We applied PARSE to two tasks, film recommendation and privacy bias evaluation, across three models (GPT-4o-mini, Llama 3.2, and DeepSeek-7B). Our results converge under the finding that the linguistic form of a prompt, independent of its semantic content, systematically biases LLM outputs, and the magnitude of that bias scales with the extent of token perturbations. The results across both case studies confirm that grammatical, typographical, and dialectal perturbations each shift LLM outputs, that

the magnitude of shift scales with the extent of textual change, and that the direction of shift is consistent across models while magnitude varies.

Building on the current evaluation, the PARSE framework can be extended along several dimensions. First, varying typographical noise intensity across multiple levels (e.g., 5%, 15%, 30% of words) would test whether the relationship between noise level and output shift is linear or exhibits threshold effects. Second, extending the evaluation to additional prompt templates that admit a wider range of the 189 grammatical features would clarify whether the limited grammatical effects observed here are specific to the features tested or reflect a general robustness of LLMs to grammatical variation. On the task side, applying PARSE to additional domains such as healthcare advice, legal guidance, and job recommendation would test the generalizability of the observed patterns, the CI evaluation could be extended beyond the educational domain to healthcare or employment vignettes. On the model side, evaluating larger open-weight models and additional commercial APIs would clarify how model scale and alignment method interact with perturbation sensitivity. Finally, across both case studies, we observe that grammatical perturbations are more likely to affect model outputs when they cover a larger portion of the prompt, suggesting that perturbation effects could be normalized by the fraction of tokens edited relative to total prompt length to support a unified measure of linguistic robustness across tasks.

Bibliography

[1] Federico Errica, Giuseppe Siracusano, Davide Sanvito, and Roberto Bifulco. What did i do wrong? quantifying llms' sensitivity and consistency to prompt engineering. In *Proceedings of the 2025 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 1543–1558, 2025. doi: 10.18653/v1/2025.naacl-long.73. URL `https://aclanthology.org/2025.naacl-long.73`. Prompt sensitivity and consistency in LLM outputs.

[2] Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Margaret Mitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency (FAccT)*, pages 610–623, Virtual Event, 2021. ACM. doi: 10.1145/3442188.3445922. URL `https://dl.acm.org/doi/10.1145/3442188.3445922`. Influential essay on large LMs' risks (bias, misinformation, etc.).

[3] OpenAI. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. doi: 10.48550/arXiv.2303.08774. URL `https://arxiv.org/abs/2303.08774`. OpenAI report detailing GPT-4's capabilities and limitations.

[4] Dan Milmo. Man develops rare condition after chatgpt query over stopping eating salt. The Guardian (Technology News), August 12 2025. URL `https://www.theguardian.com/technology/2025/aug/12/`

us-man-bromism-salt-diet-chatgpt-openai-health-information. News article on health harm from AI advice.

[5] Robert Booth. Teen killed himself after 'months of encouragement from chatgpt', lawsuit claims. The Guardian (Technology), August 27 2025. URL https://www.theguardian.com/technology/2025/aug/27/chatgpt-scrutiny-family-teen-killed-himself-sue-open-ai. News article on lawsuit alleging chatbot's role in teen's suicide.

[6] Caleb Ziems, William Held, Jingfeng Yang, Jwala Dhamala, Rahul Gupta, and Diyi Yang. Multi-VALUE: A framework for cross-dialectal English NLP. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 744–768, Toronto, Canada, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.44. URL https://aclanthology.org/2023.acl-long.44/.

[7] Ankit Kumar, Piyush Makhija, and Anuj Gupta. Noisy Text Data: Achilles' Heel of BERT. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 16–21, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.wnut-1.3. URL https://www.aclweb.org/anthology/2020.wnut-1.3.

[8] Maarten Sap, Dallas Card, Saadia Gabriel, Yejin Choi, and Noah A. Smith. The risk of racial bias in hate speech detection. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1668–1678, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1163. URL https://aclanthology.org/P19-1163/.

[9] Joel Mire, Zubin Trivadi Aysola, Daniel Chechelnitsky, Nicholas Deas, Chrysoula Zerva, and Maarten Sap. Rejected dialects: Biases against African American language in reward models. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 7483–7502, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.findings-naacl.417. URL https://aclanthology.org/2025.findings-naacl.417/.

[10] Michael Joseph Ryan, William B. Held, and Diyi Yang. Unintended impacts of llm alignment on global representation. In *Annual Meeting of the Association for Computational Linguistics*, 2024. URL https://api.semanticscholar.org/CorpusID:267897555.

[11] Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.442. URL https://aclanthology.org/2020.acl-main.442/.

[12] Moran Mizrahi, Guy Kaplan, Dan Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. State of what art? a call for multi-prompt LLM evaluation. *Transactions of the Association for Computational Linguistics*, 12:933–949, 2024. doi: 10.1162/tacl_a_00681. URL https://aclanthology.org/2024.tacl-1.52/.

[13] Yan Shvartzshnaider and Vasisht Duddu. Privacy bias in language models: A contextual integrity-based auditing metric, 2025. URL https://arxiv.org/abs/2409.03735.

[14] Fred J. Damerau. A technique for computer detection and correction of spelling errors. *Communications of the ACM*, 7(3):171–176, 1964. doi: 10.1145/363958.363994.

[15] Vladimir I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710, 1966. URL https://nymity.ch/sybilhunting/pdf/Levenshtein1966a.pdf.

[16] Helen Nissenbaum. *Privacy in Context*. Stanford University Press, Redwood City, 2009. ISBN 9780804772891. doi: doi:10.1515/9780804772891. URL https://doi.org/10.1515/9780804772891.

[17] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. URL https://arxiv.org/abs/2203.02155.

[18] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL https://arxiv.org/abs/2305.18290.

[19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 4171–4186. Association for Computational Linguistics, 2019. URL https://aclanthology.org/N19-1423.

[20] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683, 2019. URL http://arxiv.org/abs/1910.10683.

[21] Bernd Kortmann, Kerstin Lunkenheimer, and Katharina Ehret, editors. *eWAVE*. 2020. URL https://ewave-atlas.org/.

[22] Nicholas Deas, Jessica Grieser, Shana Kleiner, Desmond Patton, Elsbeth Turcan, and Kathleen McKeown. Evaluation of African American language bias in natural language generation. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6805–6824, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.421. URL https://aclanthology.org/2023.emnlp-main.421/.

[23] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *CoRR*, abs/1508.07909, 2015. URL http://arxiv.org/abs/1508.07909.

[24] Panuthep Tasawong, Wuttikorn Ponwitayarat, Peerat Limkonchotiwat, Can Udomcharoenchaikit, Ekapol Chuangsuwanich, and Sarana Nutanong. Typo-robust representation learning for dense retrieval. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1082–1090, Toronto, Canada, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-short.95. URL https://aclanthology.org/2023.acl-short.95/.

[25] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. ByT5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306, 2022. doi: 10.1162/tacl_a_00461. URL https://aclanthology.org/2022.tacl-1.17/.

[26] Hongyi Zheng and Abulhair Saparov. Noisy Exemplars Make Large Language Models More Robust: A Domain-Agnostic Behavioral Analysis. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4560–4568, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.277. URL https://aclanthology.org/2023.emnlp-main.277.

[27] Douwe Kiela, Max Bartolo, et al. Dynabench: Rethinking benchmarking in NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5193–5207, Online, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.324. URL https://aclanthology.org/2021.naacl-main.324/.

[28] Wenyue Hua, Yingqiang Ge, Shuyuan Xu, Jianchao Ji, and Yongfeng Zhang. Up5: Unbiased foundation model for fairness-aware recommendation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1899–1912, St. Julian's, Malta, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.eacl-long.114. URL https://aclanthology.org/2024.eacl-long.114/.

[29] Shijie Wang, Wenqi Fan, Yue Feng, Lin Shanru, Xinyu Ma, Shuaiqiang Wang, and Dawei Yin. Knowledge graph retrieval-augmented generation for LLM-based recommendation. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors,

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 27152–27168, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1317. URL https://aclanthology.org/2025.acl-long.1317/.

[30] Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian McAuley, and Wayne Xin Zhao. Large language models are zero-shot rankers for recommender systems. In *Advances in Information Retrieval: 46th European Conference on Information Retrieval (ECIR)*, volume 14609 of *Lecture Notes in Computer Science*, pages 364–381. Springer, 2024. doi: 10.1007/978-3-031-56060-6_24.

[31] Sunhao Dai, Ninglu Shao, Haiyuan Zhao, Weijie Yu, Zihua Si, Chen Xu, Zhongxiang Sun, Xiao Zhang, and Jun Xu. Uncovering ChatGPT's capabilities in recommender systems. In *Proceedings of the 17th ACM Conference on Recommender Systems (RecSys '23)*, pages 1126–1132. ACM, 2023. doi: 10.1145/3604915.3610646.

[32] Asia J. Biega, Krishna P. Gummadi, and Gerhard Weikum. Equity of attention: Amortized fairness in rankings. In *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 405–414, Ann Arbor, MI, 2018. ACM. doi: 10.1145/3209978.3210063.

[33] Ashudeep Singh and Thorsten Joachims. Fairness of exposure in rankings. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2219–2228, London, UK, 2018. ACM. doi: 10.1145/3219819.3220088.

[34] Ashudeep Singh, Yoni Halpern, Nithum Thain, Konstantina Christakopoulou, Ed H. Chi, Jilin Chen, and Alex Beutel. Building healthy recommendation sequences for everyone: A safe reinforcement learning approach. 2021. URL https://api.semanticscholar.org/CorpusID:231878167.

[35] Noah Apthorpe, Yan Shvartzshnaider, Arunesh Mathur, Dillon Reisman, and Nick Feamster. Discovering smart home internet of things privacy norms using contextual integrity. In *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, volume 2, pages 1–23, 2018. doi: 10.1145/3214262.

[36] Yan Shvartzshnaider, Schrasing Tong, Thomas Wies, Paula Kift, Helen Nissenbaum, Lakshminarayanan Subramanian, and Prateek Mittal. Learning privacy expectations by crowdsourcing contextual informational norms. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 4, pages 209–218, 2016. doi: 10.1609/hcomp.v4i1.13271.

[37] Niloofar Mireshghallah, Hyunwoo Kim, Xuhui Zhou, Yulia Tsvetkov, Maarten Sap, Reza Shokri, and Yejin Choi. Can llms keep a secret? testing privacy implications of language models via contextual integrity theory, 2024. URL https://arxiv.org/abs/2310.17884.

[38] Esther Gan, Yiran Zhao, Liying Cheng, Yancan Mao, Anirudh Goyal, Kenji Kawaguchi, Min-Yen Kan, and Michael Shieh. Reasoning Robustness of LLMs to Adversarial Typographical Errors, November 2024. URL http://arxiv.org/abs/2411.05345. arXiv:2411.05345 [cs].

[39] Xiao Wang, Qin Liu, Tao Gui, Qi Zhang, Yicheng Zou, Xin Zhou, Jiacheng Ye, Yongxin Zhang, Rui Zheng, Zexiong Pang, Qinzhuo Wu, Zhengyan Li, Chong Zhang, Ruotian Ma, Zichu Fei, Ruijian Cai, Jun Zhao, Xingwu Hu, Zhiheng Yan, Yiding Tan, Yuan Hu, Qiyuan Bian, Zhihua Liu, Shan Qin, Bolin Zhu, Xiaoyu Xing, Jinlan Fu, Yue Zhang, Minlong Peng, Xiaoqing Zheng, Yaqian Zhou, Zhongyu Wei, Xipeng Qiu, and Xuanjing Huang. Textflint: Unified multilingual robustness evaluation toolkit for natural language processing. In Heng Ji, Jong C. Park, and Rui Xia, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 347–355, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-demo.41. URL https://aclanthology.org/2021.acl-demo.41/.

[40] Graham E. Rawlinson. *The Significance of Letter Position in Word Recognition*. PhD thesis, University of Nottingham, 1976.

[41] BerriAI. Litellm: Call 100+ llm apis in openai format, 2024. URL https://github.com/BerriAI/litellm. Accessed: 2026-02-25.

[42] Shuyin Ouyang, Jie M. Zhang, Mark Harman, and Meng Wang. An empirical study of the non-determinism of chatgpt in code generation. *ACM Trans. Softw. Eng. Methodol.*, 34(2), January 2025. ISSN 1049-331X. doi: 10.1145/3697010. URL https://doi.org/10.1145/3697010.

[43] Berk Atil, Sarp Aykent, Alexa Chittams, Lisheng Fu, Rebecca J. Passonneau, Evan Radcliffe, Guru Rajan Rajagopal, Adam Sloan, Tomasz Tudrej, Ferhan Ture, Zhe Wu, Lixinyu Xu, and Breck Baldwin. Non-determinism of "deterministic" llm settings, 2025. URL https://arxiv.org/abs/2408.04667.

[44] OpenAI. Gpt-4o mini (2024-07-18) via the openai chat completions api. Online API documentation, 2024. URL https://platform.openai.com/docs/models/gpt-4o-mini. Model accessed via OpenAI Chat Completions API. URL: https://platform.openai.com/docs/models/gpt-4o-mini.

[45] Meta / Meta Llama Team. Llama 3.2 instruct served with llama.cpp. Model card and open-source implementation, 2024. URL https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct. Instruction-tuned model available via Hugging Face and other sources. URL: https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct.

[46] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. URL https://arxiv.org/abs/2407.21783.

[47] Xiao Bi, Deli Chen, Guanting Chen, et al. DeepSeek LLM: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024. URL https://arxiv.org/abs/2401.02954.

[48] sigpwned. Popular names by country dataset. https://github.com/sigpwned/popular-names-by-country-dataset, 2023. Accessed: 2025-09-28.

[49] William H. Greene. *Econometric Analysis*. Pearson, 8 edition, 2018.

[50] Teng Wang, Zhenqi He, Wing-Yin Yu, Xiaojin Fu, and Xiongwei Han. Large language models are good multi-lingual learners : When LLMs meet cross-lingual prompts. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert, editors, *Proceedings of the 31st International Conference on Computational Linguistics*, pages 4442–4456, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.coling-main.300/.

Appendices

A Full List of PARSE features

Table 17: Full List of PARSE Grammar Features

Name	Example (standard)	Example (variant)
A-prefix -ing	Where are you going?	Where are you a-goin?
A-prefix participle	You've killed your mother.	You've a-killed your mother.
Absolute reflex himself	and he and the bull were tuggin' and wrestlin'	and himself and the bull were tuggin' and wrestlin'
Acomp focusing like	It was really cheap.	It was like really cheap.
Adjective postfix	A big and fresh fish is my favorite.	A fish big and fresh is my favorite.
After perfect	She has just sold the boat.	She's after selling the boat.
Ain't for be	She is not going.	She ain't going.
Ain't before main verb	He did not tell me.	He ain't tell me.
Ain't for have	I have not seen it.	I ain't seen it.
Analytic superlative	one of the prettiest sunsets	one of the most pretty sunsets
Analytic whose relativizer	This is the man whose wife has died.	This is the man that his wife has died.
Anaphoric it	Things have become more expensive than they used to be.	Things have become more expensive than it used to be.
Ass pronoun	That car is fast.	That car is ass fast.
Bare ccomp	When mistress started whooping her, she sat her down.	When mistress started whoop her, she sat her down.
Bare past tense	They came and joined us.	They come and joined us.
Bare perfect	We had caught the fish when the big wave hit.	We had catch the fish when the big wave hit.
Be perfect	They haven't left school yet.	They're not left school yet.
Benefactive dative	I have to get one of those!	I have to get me one of those!
Chaining main verbs	If you stay longer, they have to charge more.	Stay longer, they have to over-charge.
Clause final really/but	I don't know what else she can do, really.	I don't know what else she can do, but.
Clause final though/but	There's nothing wrong with this box though.	There's nothing wrong with this box, but.
Clefting	A lot of them are looking for more land.	It's looking for more land a lot of them are.

Continued on next page

Continued from previous page

Name	Example (standard)	Example (variant)
Come future	I am about to cook your meal.	I am coming to cook your meal.
Comparative as to	She is bigger than her sister.	She is bigger as her sister.
Comparative more and	He has more clothes than all of us.	He has more clothes and all of us.
Comparative than	They like football more than basketball.	They like football than basketball.
Completive done	She has already left	She done left
Completive finish	I have eaten.	I finish eat.
Completive have done	He has talked about me.	He has done talked about me.
Conditional were/was	If I were you I would go home now.	If I was you I would go home now.
Correlative conjunction doubling	Despite being instructed on what to do, he still made some mistakes.	Despite being instructed on what to do still yet he made some mistakes.
Correlative constructions	The ones I made are the good ones.	The one I made, that one is good.
Definite with abstract	I stayed on until Christmas.	I stayed on until the Christmas.
Definite for indefinite	She's got a toothache.	She's got the toothache
Degree adj for adv	That's really nice and cold	That's real nice and cold
Demonstrative for definite	They have two children. The elder girl is 19 years old.	They have two children. That elder girl is 19 years old.
Demonstrative no number	These books are useful for my study.	This books are useful for my study.
Do tense marker	I knew some things weren't right.	I did know some things weren't right.
Invariant don't	He doesn't always tell the truth	He don't always tell the truth
Double comparative	That is so much easier to follow.	That is so much more easier to follow.
Double determiners	This common problem of ours is very serious.	This our common problem is very serious.
Double modals	We could do that.	We might could do that.
Double obj order	She would teach it to us.	She'd teach us it.
Double past	They didn't make it this time.	They didn't made it this time.
Double superlative	He is the most regular guy I know.	He is the most most regular guy I know.
Doubly filled comp	Who ate what?	What who has eaten?

Continued on next page

Continued from previous page

Name	Example (standard)	Example (variant)
Drop aux be gonna	He is gonna go home and watch TV.	He gonna go home and watch TV.
Drop aux be progressive	You are always thinking about it.	You always thinking about it.
Drop aux have	I have seen it before.	I seen it before.
Drop aux wh	When is she coming?	When she coming?
Drop aux yn	Do you get the point?	You get the point?
Drop copula be AP	She is smart.	She smart.
Drop copula be NP	He is a good teacher.	He a good teacher.
Drop copula be locative	She is at home.	She at home.
Drop inf to	They were allowed to call her.	They were allowed call her.
'em object pronoun	We just turned it around.	We just turned 'im around.
'em subject pronoun	This old woman, she started packing up.	This old woman, 'em started packing up.
Emphatic reflex their own self	They brought it by themselves.	They brought it by their own self.
Existential got	There's no water in the toilet.	Got no water in the toilet.
Existential it	There is a problem	It's a problem
Existential possessives	I have a son.	Son is there.
Existential there (singular agreement)	There are two men waiting	There's two men waiting
Existential you have	There are some people who don't give a damn about animals.	You have some people they don't give a damn about animals.
Finna/fixin future	They're about to leave.	They're fixin to leave town.
Fixin future	They're going to leave.	They're fixin to leave.
Flat adj for adv	She speaks so softly.	She speaks so soft.
For complementizer	You mean your mother allows you to bring over boyfriends?	You mean your mother allows you for bring over boyfriends?
For to	He had the privilege to turn on the lights.	He had the privilege for to turn on the lights.
For to purpose	We always had gutters in the winter time to drain the water away.	We always had gutters in the winter time for to drain the water away.
Fronting pobj	I drive to town every Saturday.	To town every Saturday I drive.
Future sub gon	He will come with us.	He gon' come with us.
Generalized third person -s	Every Sunday we go to church.	Every Sunday we goes to church.

Continued on next page

Continued from previous page

Name	**Example (standard)**	**Example (variant)**
Give passive	John was scolded by his boss	John give his boss scold.
Got for have	I have a car.	I got a car.
Got → gotten	I hope you've got your topic already.	I hope you've gotten your topic already.
He for inanimate	The driver's license? She wasn't allowed to renew it right?	The driver's license? She wasn't allowed to renew 'im right?
Here come	Bring the book here.	Take the book bring come.
His → he (possessive)	his book	he book
His → him (possessive)	his book	him book
If would	If I were you I would go home now.	If I would be you I would go home now.
Indefinite one	What happened? Oh, a dog bit me.	What happened? Oh, one dog bit me.
Indefinite for definite	The moon was very bright last night.	A moon was very bright last night.
Indefinite for zero	We received good news at last.	We received a good news at last.
Invariant tag amn't I	I believe I am older than you. Is that correct?	I am older than you, amn't I?
Invariant tag can or not	Can I go home?	I want to go home, can or not?
Invariant tag fronted isn't	I can go there now can't I?	Isn't, I can go there now?
Invariant tag non-concord	I believe you are ill. Is that correct?	You are ill, isn't it?
Inverted indirect question	I'm wondering what you are going to do.	I'm wondering what are you going to do.
Irrealis be done	If you love your enemies, they will eat you alive in this society.	If you love your enemies, they be done eat you alive in this society.
Is/am 1s	I am going to town.	I's going to town.
It as direct object	As I explained to her, this is not the right way.	As I explained it to her, this is not the right way.
It is non-referential drop	Okay, it's time for lunch.	Okay, is time for lunch.
It is referential drop	It is very nice food.	Is very nice food.

Continued on next page

Continued from previous page

Name	**Example (standard)**	**Example (variant)**
Linking relative clause	Unless you are going to get 88, but some universities are not going to give those marks	Unless you are going to get 88 which some universities are not going to give those marks
Mass noun plurals	furniture, machinery, equipment, evidence, luggage, advice, mail, staff	furnitures, machineries, equipments, evidences, luggages, advices, mails, staffs
Me in coordinate subjects	Michelle and I will come too.	Me and Michelle will come too.
Me → us (object)	Show me the town!	Show us the town!
Medial object perfect	He has written a letter.	He has a letter written.
More → much	The situation is more serious than I thought.	The situation is much serious than I thought.
My → I (possessive)	my book	I book
My → me (possessive)	my book	me book
Myself in coordinate subjects	My husband and I were late.	My husband and myself were late.
Nasal possessive pronoun	her, his, our; hers, ours, ours	hern, hisn, ourn; hersn, oursn, ourns
Negative concord	I don't have anything	I don't have nothing
Negative inversion	Nobody showed up.	Didn't nobody show up.
Never as negator	He didn't come	He never came
No gender distinction	Susan is a nurse but she does not like to put drips on patients.	Susan is a nurse but he does not like to put drips on patients.
No preverbal negator	I don't want any job or anything.	I no want any job or anything.
Nomo existential	There is not any food in the refrigerator.	No more food in the refrigerator.
Non-coordinated obj/subj them	They can ride all day.	Them can ride all day.
Non-coordinated subj/obj with we	Do you want to come with us?	Do you want to come with we?
Not preverbal negator	The baby didn't eat food and cried a lot.	The baby not ate food and cried a lot.
Null genitive	my cousin's bike	my cousin bike
Null prepositions	I'm going to town.	I'm going town.
Null referential pronouns	When I come back from my work I just travel back to my home.	When I come back from my work just travel back to my home.

Continued on next page

Continued from previous page

Name	**Example (standard)**	**Example (variant)**
Null relative clause	The man who lives there is friendly.	The man lives there is friendly.
Object pronoun drop	I got it from the store.	I got from the store.
One relativizer	The cake that John buys is always very nice to eat.	The cake John buy one always very nice to eat.
Our → us (possessive)	our book	us book
Our → we (possessive)	our farm	we farm
Participle past tense	I saw it.	I seen it.
Past been	I told you.	I been told you.
Past for past participle	He had gone.	He had went.
Perfect already	Have you eaten lunch?	Did you eat already?
Perfect slam	I have already told you not to mess up	I slam told you not to mess up.
Pleonastic that	It's raining.	Thass raining.
Plural interrogative who-all	Who came?	Who-all came?
Plural to singular human	The three girls there don't want to talk to us.	The three girl there don't want to talk to us.
Possessives belong	the woman's friend	woman belong friend
Possessives for post	This is my mother's house.	This is the house for my mother.
Possessives for pre	Long time ago he was my sister's husband.	Long time he was for my sister husband.
Preposition chopping	You remember the swing that we all used to sit together on?	You remember the swing that we all used to sit together?
Present for experiential perfect	I've known her since she was a child.	I know her since she was a child.
Present for neutral future	Next week, I will be leaving the States and going to Liberia.	Next week, I leaving the States, I going to Liberia.
Present modals	I wish I could get the job.	I wish I can get the job.
Present perfect ever	I have seen the movie.	I ever see the movie.
Present perfect for past	We were there last year.	We've been there last year.
Progressives	I like her hair style right now.	I am liking her hair style.
Proximal distal demonstratives	this book that is right here vs. those books that are over there	this here book vs. them there books

Continued on next page

Continued from previous page

Name	**Example (standard)**	**Example (variant)**
Quotative like	And my friend said "No way!"	And my friend was like "No way!"
Reduced relative	There is nothing like food cooked by Amma!	There is nothing like Amma cooked food!
Reduplicate interrogative	Who's coming today?	Who-who's coming today?
Referential 'the thing'	Christmas dinner? I think it's better to wait until after she's had it.	Christmas dinner? I think it's better to wait until after she's had the thing.
Reflexive number ourself	We cannot change ourselves.	We cannot change ourself.
Regularized past tense	He caught the ball.	He catched the ball.
Regularized plurals	wives, knives, lives, leaves	wifes, knifes, lifes, leafs
Regularized reflexives hisself	He hurt himself.	He hurt hisself.
Regularized reflexives theyselves	They look after themselves.	They look after theyselves.
Regularized reflexives meself	I'll do it myself.	I'll do it meself.
Relativizer doubling	But these, these little fellahs who had stayed before	But these, these little fellahs that which had stayed befo'
Relativizer where	My father was one of the founders of the Underground Railroad, which helped the slaves to run away to the North	My father was one o de founders o' de Underground Railroad where help de slaves to run way to de North.
Remove definite article	He's in the office.	He's in office.
Remove indefinite article	Can I get a better grade?	Can I get better grade?
Say complementizer	We hear that you were gone to the city.	We hear say you gone to the city.
Serial verb give	I bought rice for you.	I buy rice give you.
Serial verb go	Grandfather sends us to school.	Grandfather send us go school.
Shadow pronouns	This is the house which I painted yesterday.	This is the house which I painted it yesterday.
She for inanimate	It's a good bike	She's a good bike
Simple past for present perfect	I've eaten the food. So can I go now?	I ate the food. So can I go now?

Continued on next page

Continued from previous page

Name	**Example (standard)**	**Example (variant)**
Standing → stood	He was standing on the corner.	He was stood on the corner.
Subord conjunction doubling	Although you are smart, you are not appreciated	Although you are smart, but you are not appreciated
Superlative before matrix head	The thing I like most is apples.	The most thing I like is apples.
Synthetic superlative	He is the most regular guy I know.	He is the regularest guy I know.
That infinitival subclause	He wanted me to go with him.	He wanted that I should go with him.
That resultative past participle	There is a car that broke down on the road.	There is a car broken down on the road.
Their → them (possessive)	their book	them book
Their → they (possessive)	their book	they book
Those → them (demonstrative)	I don't have any of those qualifications.	I don't have any of them qualifications.
To infinitive	He made me do it.	He made me to do it.
Too sub	They are very nice. We had a good time there.	They are too nice. We had a good time there.
Transitive suffix	You can see the fish.	You can see 'im fish.
Uninflect	He speaks English.	He speak English.
Verbal -ing suffix	I can drive now.	I can driving now.
Volition changes	You want to go.	You waan go.
Wasn't/weren't leveling	John was there, but Mike wasn't	John was there, but Mike weren't
Were/was leveling	You were hungry but he was thirsty.	You was hungry but he was thirsty.
What comparative	I'm taller than he is.	I'm taller than what he is.
Who/as	The man who was just here.	The man as was just here.
Who/at	This is the man who painted my house.	This is the man at painted my house.
Who/what	This is the man who painted my house.	This is the man what painted my house.
Who/which	He's the man who looks after the cows.	He's the man which looks after the cows.
Will → would	I will meet him tomorrow.	I would meet him tomorrow.
Y'all	you	y'all
You → ye	Sure it's no good to you in England.	Sure it's no good to ye in England.
Your → y'all's	Where are your books?	Where are y'all's books?

Continued on next page

Continued from previous page

Name	Example (standard)	Example (variant)
Your → you (possessive)	your book	you book
Zero degree	He is one of the most radical students that you can ever find.	He is one of the radical students that you can ever find.
Zero plural	Some apartments are bigger.	Some apartment are bigger.
Zero plural after quantifier	It's only five miles away.	It's only five mile away.

B CI Parameter Translation Dictionary

Table 18: CI Parameter Translation Dictionary

Category	Original term	Gen Z	AAVE	Indian English
Attributes				
Attribute	address	addy	where you at	address
Attribute	attendance	showing up vibes	showin' up	attendance
Attribute	contents_online	online deets	stuff online	online content
Attribute	email	slid into the inbox	email	email
Attribute	grades	scorez	marks	marks
Attribute	library_record	book stash	library history	library record
Attribute	name	username	yo name	name
Attribute	participation	engagement level	gettin' involved	participation
Attribute	photo	pic	picture	photo
Attribute	term_papers	big brain essays	papers for the term	term papers
Attribute	transcript	grade sheet flex	school record	transcript
Conditions (Transmission principle realizations)				
Condition	if requested by IT_staff	if IT peeps slide into the DMs	if IT staff ask for it	if requested by IT staff
Condition	if requested by TA	if the TA hits me up	if the TA ask for it	if requested by TA
Condition	if requested by advisor	if my advisor asks for it	if the advisor ask for it	if requested by advisor
Condition	if requested by classmates	if my squad asks for it	if the classmates ask for it	if requested by batchmates
Condition	if requested by department_chair	if the department head comes knocking	if the department chair ask for it	if requested by department head
Condition	if requested by graduate_schools	if grad schools wanna know	if the grad schools ask for it	if requested by graduate schools
Condition	if requested by librarian	if the library homie asks	if the librarian ask for it	if requested by librarian

Continued on next page

Category	Original term	Gen Z	AAVE	Indian English
Condition	if requested by parents	if the 'rents wanna know	if the parents ask for it	if requested by parents
Condition	if requested by professor	if the prof is curious	if the professor ask for it	if requested by professor
Condition	if requested by registrar	if the registrar needs it	if the registrar ask for it	if requested by registrar
Condition	if the student is performing poorly	if the student is kinda struggling	if the student ain't doin' good	if the student is performing poorly
Condition	if the student's IT_staff asked for the student's permission	if IT peeps got the green light from the student	if the IT staff asked the student if it's cool	if the student's IT staff asked for the student's permission
Condition	if the student's IT_staff let the student know	if IT peeps gave the student a heads up	if the IT staff told the student	if the student's IT staff informed the student
Condition	if the student's TA asked for the student's permission	if the TA got the student's okay	if the TA asked the student if it's cool	if the student's TA asked for the student's permission
Condition	if the student's TA let the student know	if the TA filled the student in	if the TA told the student	if the student's TA informed the student
Condition	if the student's advisor asked for the student's permission	if the advisor got the student's thumbs up	if the advisor asked the student if it's cool	if the student's advisor asked for the student's permission
Condition	if the student's advisor let the student know	if the advisor dropped the info to the student	if the advisor told the student	if the student's advisor informed the student
Condition	if the student's classmates asked for the student's permission	if the squad got the student's go-ahead	if the classmates asked the student if it's cool	if the student's batchmates asked for the student's permission
Condition	if the student's classmates let the student know	if the squad kept the student in the loop	if the classmates told the student	if the student's batchmates informed the student
Condition	if the student's librarian asked for the student's permission	if the library homie got the student's okay	if the librarian asked the student if it's cool	if the student's librarian asked for the student's permission
Condition	if the student's librarian let the student know	if the library homie gave the student the 411	if the librarian told the student	if the student's librarian informed the student
Condition	if the student's professor asked for the student's permission	if the prof got the student's blessing	if the professor asked the student if it's cool	if the student's professor asked for the student's permission
Condition	if the student's professor let the student know	if the prof clued the student in	if the professor told the student	if the student's professor informed the student

Continued on next page

Category	Original term	Gen Z	AAVE	Indian English
Condition	if the student's registrar asked for the student's permission	if the registrar got the student's nod	if the registrar asked the student if it's cool	if the student's registrar asked for the student's permission
Condition	if the student's registrar let the student know	if the registrar kept the student posted	if the registrar told the student	if the student's registrar informed the student
Condition	with the requirement of confidentiality	with the need to keep it on the down-low	gotta keep it on the low	with the requirement of confidentiality
Recipients				
Recipient	IT staff	tech wizards	tech folks	IT staff
Recipient	TA	teaching homies	teaching assistant	teaching assistant
Recipient	advisor	life coach	mentor	mentor
Recipient	classmates	study squad	peers	batchmates
Recipient	department chair	head honcho	head of the department	head of department
Recipient	graduate schools	grad vibes	grad schools	postgraduate institutes
Recipient	librarian	book guru	library person	librarian
Recipient	parents	fam squad	folks	parents
Recipient	professor	knowledge dropper	prof	professor
Recipient	registrar	paperwork boss	registration office	registrar
Senders				
Sender	IT_staff	tech wizards	tech folks	IT staff
Sender	TA	teaching homie	teaching assistant	teaching assistant
Sender	advisor	life coach	counselor	mentor
Sender	classmates	study squad	peers	batchmates
Sender	librarian	book guru	library person	librarian
Sender	professor	knowledge dropper	prof	professor
Sender	registrar	paperwork boss	registration person	registrar
Subjects				
Subject	student	scholar vibin'	student	vidyarthi