ZIP: Quantifying Which Words Matter in Zero-Shot Instructional Prompts

Nikta Gohari Sadr[1,2], **Sangmitra Madhusudan**[1], **Arash Asgari**[2,3], **Hassan Sajjad**[4], **Laleh Seyyed-Kalantari**[2,3†], and **Ali Emami**[5†]
[1]Brock University, [2]York University, [3]Vector Institute, [4]Dalhousie University
[5]Emory University

Abstract

While zero-shot instructional prompts like "Let's think step-by-step" have revolutionized Large Language Model performance, we lack systematic understanding of *why*: which specific words drive their effectiveness, and how do these patterns vary across tasks and models? We introduce the ZIP score (Zero-shot Importance of Perturbation), a metric that quantifies individual word importance through controlled, semantically meaningful perturbations. To enable rigorous evaluation, we also introduce the first ground-truth benchmark for prompt interpretability, a set of validation prompts with predetermined keywords where ZIP achieves 95.8% accuracy compared to 65.8% for LIME. Analyzing six flagship models across seven prompts and multiple task domains, we find that word importance is task-dependent ("step-by-step" dominates mathematical reasoning; "think" matters more for common-sense tasks), varies systematically across model families, and correlates inversely with model performance, suggesting prompts have greatest impact on tasks where models struggle. Our findings advance *prompt science*, providing both practical guidance for prompt engineering and theoretical understanding of how instructional language shapes model behavior.[1]

1 Introduction

Chain-of-Thought prompting transformed how we use Large Language Models. When Wei et al. (2022) added "Let's think step-by-step" to prompts, GPT-3 suddenly solved complex math problems it had previously failed. This discovery sparked a wave of instructional prompts: Self-Consistency (Wang et al., 2023b), Plan-and-Solve (Wang et al., 2023a), and others, each improving performance through carefully crafted phrases. These prompts have since become fundamental to eliciting reasoning capabilities in modern LLMs, including Large Reasoning Models like OpenAI's o3 (OpenAI, 2025).

Consider the Plan-and-Solve prompt: "Let's first understand the problem and devise a plan to solve it. Then, solve it step-by-step" (Wang et al., 2023a). This manually crafted extension of Chain-of-Thought consistently outperforms the original across benchmarks. The intuition *seems* clear: explicit planning should reduce errors. But which words actually drive this improvement? Is it the instruction to "plan"? The directive to "understand"? Or does the familiar "step-by-step" do most of the work? Despite extensive empirical validation, the design of these prompts remains largely *ad hoc*. We have no principled way to determine how individual word choices affect model performance.

Prior work has documented that models are highly sensitive to prompt structure: Sclar et al. (2023) found that formatting changes alone can swing accuracy by over 70 percentage points. Yet while structural sensitivity is well-established, the role of *individual word choices* within instructional prompts remains unexplored. This gap matters because word-level effects can be both substantial and counterintuitive. For instance, replacing "solve" with the near-synonym "work out" seems harmless, but can fundamentally alter model reasoning (see Section 6).

How should we measure word importance? Following established work on perturbation-based interpretability (Choudhary et al., 2022; Saleem et al., 2022; Feng et al., 2018), we define **word importance** as *the degree to which a word's presence, absence, or modification impacts model performance on a given task*. This definition is operational: a word is important if changing it changes outcomes. Alternative approaches exist. Gradient-based methods (Wallace et al., 2019; Yin and Neubig, 2022) and attention-based methods (Modarressi et al.,

[†]Co-senior authors.

[1]The complete codebase and documentation of language model interactions are available at github.com/niktaas/ZIP.

*Proceedings of the 15th Joint Conference on Lexical and Computational Semantics (*SEM 2026)*, pages 428–453

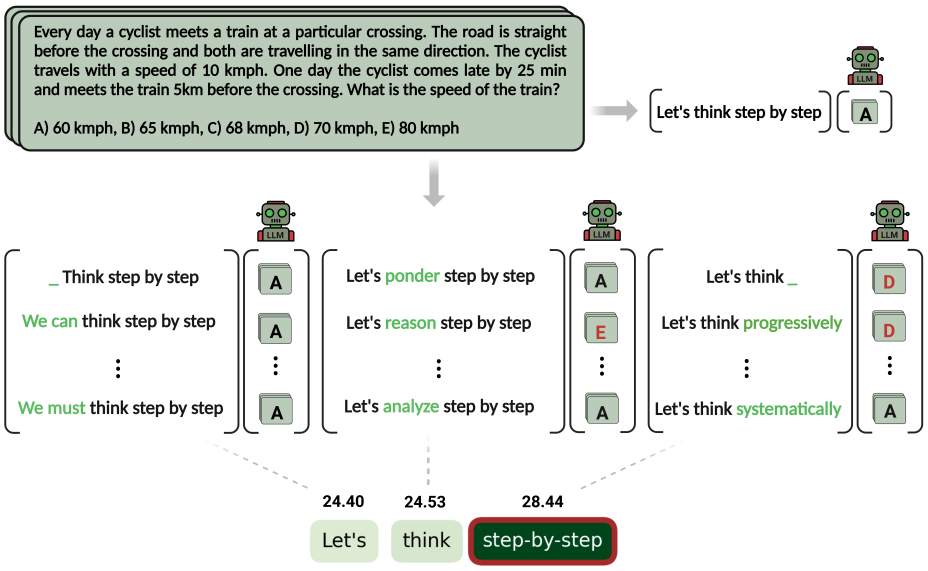


Figure 1: ZIP score generation process for the Chain-of-Thought prompt using GPT-4o on a AQUA-RAT dataset instance. Perturbed prompts are compared to the original, generating word-level ZIP scores. The red box highlights "step-by-step" as significantly important based on statistical analysis.

2022; Tenney et al., 2020) infer importance from model internals, but they measure what the model attends to or is sensitive to internally, not whether those tokens actually affect task performance.

Methods that analyze internal representations also face practical limitations. They require access to model internals, making them inapplicable to closed-source models like GPT-4. They raise reliability concerns: gradients can be manipulated (Wang et al., 2020) and attention weights often fail to correlate with actual performance impact (Jain and Wallace, 2019; Ethayarajh and Jurafsky, 2021). Even recent advances in mechanistic interpretability (Lindsey et al., 2025), while providing detailed internal analysis, require substantial computational resources and whitebox access. For understanding how word choices affect model outputs in practice, we need lightweight, model-agnostic methods that directly measure performance changes.

We introduce the ZIP score (Zero-shot Importance of Perturbation), a metric that quantifies word importance through systematic perturbations: synonym replacement, co-hyponym substitution, and word removal. ZIP works with any model, requiring only query access, making it applicable to both open-source and closed-source systems. To enable rigorous evaluation of prompt interpretability methods, we also introduce the first ground-truth benchmark: a set of validation prompts with predetermined keywords, where the correct output depends entirely on a single known word. On this benchmark, ZIP achieves 95.8% accuracy in identifying the key word, compared to 65.8% for LIME (Ribeiro et al., 2016). Figure 1 illustrates our method identifying "step-by-step" as the critical component in Chain-of-Thought prompting for a mathematical reasoning task.

Testing across six flagship models, seven prompts, and multiple task domains, our analysis reveals four key findings:

(1) **Task-specific word importance:** different words matter for different tasks, with "step-by-step" dominating mathematical reasoning while "think" matters more for common-sense tasks;

(2) **Cross-model variation:** proprietary and open-source models exhibit distinct sensitivity patterns, with GPT-4o and Llama-3 aligning more closely with human intuitions;

(3) **Syntactic regularity:** nouns consistently emerge as the most important part of speech across all models (47%–66% of significant words); and

(4) **Task difficulty effects:** ZIP scores correlate inversely with model performance ($|r| > 0.9$), indicating that prompts have greatest impact on tasks where models struggle.

These findings advance *prompt science* (Shah, 2025), the systematic study of how language shapes model behavior, by providing both practical guidance for prompt engineering and theoretical insight into when and why instructional prompts matter.

2 Related Work

Understanding which input features drive model predictions is central to interpretability research. We situate ZIP within this landscape, focusing on perturbation-based methods while contrasting with gradient-based and attention-based alternatives.

Perturbation-based methods measure feature importance by observing how model outputs change when inputs are modified (Choudhary et al., 2022). This approach is model-agnostic: it requires only query access, making it applicable to closed-source systems. Notable methods include LIME (Ribeiro et al., 2016), which fits local linear approximations, and extensions to few-shot demonstrations (Liu et al., 2023; Lu et al., 2022) and system prompts (Hackmann et al., 2024; Yin et al., 2023). However, existing perturbation approaches for prompts typically rely on simple token masking or random removal, which can produce incoherent inputs and misleading importance estimates (Yin et al., 2023; Feng et al., 2018; Kim et al., 2020). ZIP addresses this limitation through semantically meaningful perturbations (synonyms, co-hyponyms) that remain close to the original input distribution.

Gradient-based and attention-based methods infer importance from model internals. Gradient-based methods compute the gradient of output logits with respect to input elements (Wallace et al., 2019; Yin and Neubig, 2022; Ferrando et al., 2023), while attention-based methods analyze attention weight distributions (Modarressi et al., 2022; Tenney et al., 2020; Ferrando et al., 2022). Both face practical and conceptual limitations. Practically, they require whitebox access, precluding analysis of closed-source models. Conceptually, gradients can be manipulated (Wang et al., 2020), and attention weights often fail to correlate with actual performance impact (Jain and Wallace, 2019; Ethayarajh and Jurafsky, 2021; Bastings and Filippova, 2020). These methods reveal what models are internally sensitive to, but not necessarily which tokens affect task outcomes.

Prompt modification and optimization studies examine how systematic prompt changes influence model performance (Fernando et al., 2024; Agarwal et al., 2025). While focused on optimization rather than interpretability, this literature surfaces an important insight: individual prompt components play differentiated roles in shaping model behavior. ZIP builds on this insight by quantifying these differentiated roles at the word level.

Prompt science distinguishes prompt engineering (optimizing prompts for tasks) from prompt science (using prompts to discover regularities in model behavior) (Holtzman and Tan, 2025; Shah, 2025). Studies have documented prompt brittleness: Sclar et al. (2023) found that formatting changes alone can swing accuracy by over 70 percentage points. Rather than viewing this sensitivity as a flaw, we leverage it as a tool for understanding word importance. As Holtzman and Tan (2025) argue, prompt sensitivity reflects models' attempts to "infer substantial information from limited context," making it a feature for scientific investigation. ZIP provides a systematic framework for what they call "behavioral studies that use varied prompts in structured ways to confirm hypotheses."

3 The ZIP Score

3.1 Formalization of the ZIP Score

We formalize the ZIP score through five steps: prompt representation, perturbation generation, model prediction, disagreement calculation, and score computation. We illustrate each step with a running example.

Example Task (T): Determine whether a number is prime or composite.
Example Prompt (P): "Take a deep breath and work on this problem step-by-step" (Yang et al., 2024).

Step 1: Prompt Representation. Let P represent the original prompt, where $P = w_1, w_2, \ldots, w_I$ and w_i denotes the i-th word.[2]
Example:

$$P = [\text{"Take"}, \text{"a"}, \text{"deep"}, \ldots, \text{"step-by-step"}]$$

Step 2: Perturbation Generation. For each word w_i in P, we generate a set of perturbations $\{P_{i1}, P_{i2}, \ldots, P_{iJ}\}$ through three methods: synonym replacement, co-hyponym substitution, and word removal. Synonyms preserve approximate meaning, co-hyponyms introduce semantically related alternatives, and removal tests whether the word is necessary. Each P_{ij} represents a modified version of P where w_i has been replaced or omitted.

[2]Throughout this paper, "word" refers to any token resulting from space-based tokenization, which may include contractions or compound expressions like "step-by-step."

We use GPT-4 for context-aware generation of synonyms and co-hyponyms.[3] Following perturbation-based interpretability literature (Choudhary et al., 2022; Ivanovs et al., 2021; Saleem et al., 2022), we posit that a word's importance is reflected in the model's sensitivity to any meaningful change to that word, whether substitution or removal (see Section 6 for demonstrations).

Example: For w_4 = "breath":

- P_{41} = "Take a deep **pause** and work on ..." (synonym)
- P_{42} = "Take a deep **glimpse** and work on ..." (co-hyponym)
- P_{43} = "Take a deep **look** and work on ..." (co-hyponym)

Filtering. Not all generated perturbations are valid. As depicted in Figure 2, we filter candidates through two stages:

1. **Semantic similarity:** We use the Universal Sentence Encoder (Cer et al., 2018) to compute similarity between P and P_{ij}, retaining only perturbations with >30% similarity. This threshold balances preserving context while allowing meaningful alterations (see Table 27 in Appendix A.7).
2. **Grammaticality:** We use GPT-4 to assess whether each P_{ij} is grammatically correct and semantically coherent.

Example: Our filtering excludes P_{42} (insufficient semantic similarity and grammatical issues), leaving P_{41} and P_{43} as valid perturbations.

This process yielded an average of 9.04 valid perturbations per word across all prompts. We ensured cost-effectiveness by filtering with the Universal Sentence Encoder before API calls; for the CoT prompt, the entire process uses fewer than 2,150 tokens.[4]

Step 3: Model Prediction. We query the language model M with the original prompt P and each validated perturbation P_{ij}, obtaining predictions for a given task instance t.

Example (t): Is 29 prime or composite?

$$\text{pred}_t^M(P) = \text{"Prime"} \quad \text{(Correct)}$$
$$\text{pred}_t^M(P_{41}) = \text{"Prime"} \quad \text{(Correct)}$$
$$\text{pred}_t^M(P_{43}) = \text{"Composite"} \quad \text{(Incorrect)}$$

[3]GPT-4 consistently outperformed traditional tools like NLTK (Bird et al., 2009) and smaller models in generating contextually appropriate alternatives.

[4]Detailed prompts and examples for perturbation generation are provided in Appendix Sections A.6 and A.7.

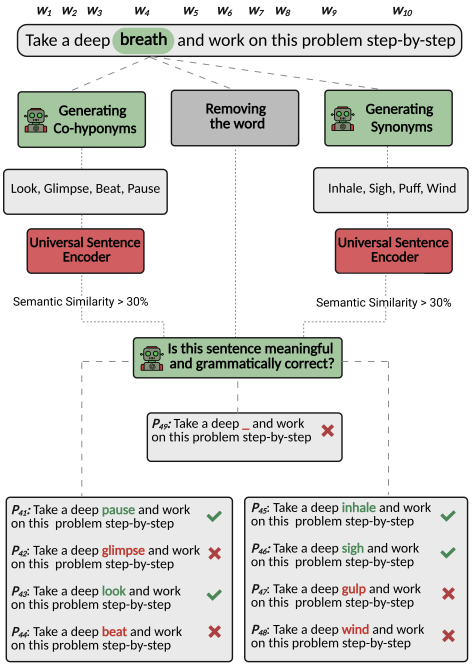


Figure 2: Perturbation generation and filtering pipeline

Step 4: Disagreement Calculation. We define a disagreement function d that measures the difference between predictions from the original prompt P and each perturbed prompt P_{ij}. The function varies by task type.

For *classification tasks*, disagreement is binary:

$$d_t^M(P, P_{ij}) = \begin{cases} 1 & \text{if } \text{pred}_t^M(P) \neq \text{pred}_t^M(P_{ij}) \\ 0 & \text{otherwise} \end{cases}$$

For *translation tasks*, disagreement is the absolute difference in BLEU scores (Papineni et al., 2002):

$$d_t^M(P, P_{ij}) = |\text{BLEU}_t^M(P) - \text{BLEU}_t^M(P_{ij})|$$

Example: Since our running example is a classification task:

$$d_t^M(P, P_{41}) = 0 \quad \text{(No disagreement)}$$
$$d_t^M(P, P_{43}) = 1 \quad \text{(Disagreement)}$$

Step 5: ZIP Score Calculation. Finally, for each word w_i, we compute $\text{ZIP}_T^M(w_i)$ by averaging disagreement scores across all J perturbations of w_i and all N dataset instances:

$$\text{ZIP}_T^M(w_i) = 100 \cdot \frac{1}{N} \sum_{t=1}^{N} \left(\frac{1}{J} \sum_{j=1}^{J} d_t^M(P, P_{ij}) \right)$$

The resulting score lies in $[0, 100]$ and represents the average percentage of cases where perturbing

word w_i changes the model's output. A higher ZIP score shows greater word importance: changes to that word more frequently alter model behavior.

Example: For w_4 = "breath" with two valid perturbations ($J = 2$) on a single instance ($N = 1$):

$$\text{ZIP}_T^M(\text{"breath"}) = 100 \cdot \frac{1}{1} \cdot \frac{1}{2}(0 + 1) = 50$$

This score of 50 indicates that perturbing "breath" changed the model's output in 50% of cases.

3.2 Identifying Significantly Important Words

Raw ZIP scores reflect average disagreement, but models exhibit inherent output variability even when prompted identically. To distinguish true perturbation effects from this baseline variability, we apply statistical testing. For each word w_i, we proceed as follows:

1. **Collect perturbation outputs:** Query the model with each perturbed prompt P_{ij}, recording the outputs.
2. **Collect baseline outputs:** Query the model repeatedly with the original prompt P, matching the number of perturbation queries to ensure balanced comparison.
3. **Statistical comparison:** Apply the Wilcoxon rank-sum test to compare the disagreement scores from perturbed prompts against those from repeated original prompts.

The Wilcoxon rank-sum test assesses whether two independent samples come from the same distribution. If the perturbation-induced disagreements significantly exceed baseline variability ($p < 0.05$), we deem the word *significantly important*. This approach ensures that words flagged as important are those whose perturbations reliably affect model outputs beyond chance variation.

4 Experimental Setup

4.1 Validation Prompts

To establish ground truth for word-level importance, we created 20 validation prompts where a single predetermined keyword determines the correct output. For example, in "Say the word green," only "green" determines success: any alteration to this word should change the model's response, while other words ("Say," "the," "word") should have minimal effect.

We designed prompts with varied syntax and structure, including "Print the digits **123**," "Type the letter **X**," and "Print **carrot** with no additional text." This variety tests whether interpretability methods can identify keywords regardless of prompt phrasing. The complete set is provided in Appendix A.2.

4.2 Prompts

We evaluated seven zero-shot instructional prompts spanning different reasoning strategies: Chain-of-Thought variants, Plan-and-Solve, irrelevant information handling, and translation-specific prompts. Table 1 lists each prompt and its associated datasets. We tokenize prompts with space-based segmentation, which keeps contractions ("Let's") and compound expressions ("step-by-step") as single units.[5]

4.3 Datasets

We evaluate on datasets commonly used to test zero-shot prompting. For classification, we use GSM8K (Cobbe et al., 2021), AQUA-RAT (Ling et al., 2017), and Big Bench (Srivastava et al., 2023). For translation, we use WMT19 (Barrault et al., 2019) for German-English and Chinese-English. We sample 150 instances per dataset, balancing statistical reliability with computational cost. Table 1 shows the prompt-dataset pairings.

4.4 Models

We evaluate six models spanning proprietary and open-source systems: GPT-4o mini, GPT-3.5 Turbo (OpenAI, 2023), Gemini-2.0 Flash (Comanici et al., 2025), Llama-2-70B Chat, Llama-3-8B Instruct (Touvron et al., 2023), and Mixtral-8x7B Instruct-v0.1 (Jiang et al., 2024). All models are queried with temperature 0.5 to allow some output variability while maintaining reasonable consistency. Token usage estimates are provided in Appendix A.8.

4.5 Baselines

We compare ZIP against LIME (Ribeiro et al., 2016), a widely-used perturbation-based interpretability method. LIME represents a natural baseline as it also uses perturbations to estimate feature importance, but relies on random word removal rather than semantically meaningful substitutions. For validation prompts, we configure LIME as a binary classifier: Label 1 corresponds to the target keyword (e.g., "green"), and Label 2 to all other outputs.

[5]Dataset-specific prompt templates are in Appendix A.6.

Code	Prompt	Mutual Datasets
0-CoT	Let's think step-by-step. (Kojima et al., 2024)	BIG-bench, GSM8K, AQUA-RAT
0-CoTB	Take a deep breath and work on this problem step-by-step. (Yang et al., 2024)	AQUA-RAT, GSM8K
0-CoTR	Let's work this out in a step-by-step way to be sure we have the right answer. (Zhou et al., 2023)	GSM8K, BIG-bench
0-IRR	Feel free to ignore irrelevant information in the problem description. (Shi et al., 2023)	GSM8k
0-PS	Let's first understand the problem and devise a plan to solve it. Then, solve it step-by-step. (Plan & Solve (Wang et al., 2023a))	GSM8K, AQUA-RAT
0-DSP	Provide the translation step-by-step, then complete the sentence. (Peng et al., 2023)	WMT19 (German, Chinese)
0-DTG	Detect the error type first, then refine the translation. (Li et al., 2023)	WMT (German, Chinese)

Table 1: Zero-shot instructional prompts used in our experiments, and mutual datasets they were tested on.

Model	ZIP Accuracy	LIME Accuracy
GPT-4o	90%	60%
GPT-3.5	100%	55%
Gemini-2.0	100%	90%
Mixtral	95%	80%
Llama-2	90%	55%
Llama-3	100%	55%
Average	**95.8%**	**65.8%**

Table 2: Accuracy of ZIP and LIME in identifying target keywords across control validation prompts.

GPT-4o	Let's	think	step-by-step
GPT-3.5	Let's	think	step-by-step
Gemini-2.0	Let's	think	step-by-step
Mixtral	Let's	think	step-by-step
Llama-2	Let's	think	step-by-step
Llama-3	Let's	think	step-by-step

Table 3: ZIP score heatmaps for the CoT prompt across six models on the Big Bench dataset. Red boxes indicate significantly important words.

4.6 Human Intuitions

To compare model-derived importance with human judgments, we recruited 20 English-proficient participants. For each prompt, participants selected up to three words they considered most important for task performance. The questionnaire is shown in Appendix A.4.

We measured inter-annotator agreement using the Jaccard Index, which accommodates multiple selections per participant (unlike Fleiss' Kappa). The average Jaccard Index of 0.4714 indicates moderate agreement, reflecting genuine variation in human intuitions about word importance. This variation underscores the value of systematic, model-based evaluation. Detailed annotations are provided in Table 22 (Appendix).

5 Results

5.1 Validation Results

Table 2 compares ZIP and LIME on our validation benchmark. ZIP substantially outperforms LIME, achieving 95.8% average accuracy versus 65.8% for LIME. ZIP achieves 100% accuracy on three models (GPT-3.5, Gemini-2.0, Llama-3), demonstrating robustness across architectures.

The performance gap reflects a key methodological difference. LIME relies on random word removal and often assigns highest importance to generic instructional tokens (e.g., "Output," "Say") rather than the keyword that determines the output. ZIP's semantically meaningful perturbations more reliably identify the word that actually matters. These results validate ZIP as a more accurate method for quantifying word-level importance. Detailed results are provided in Appendix A.2.

5.2 Cross-Model Analysis

Table 3 presents ZIP score heatmaps for the 0-CoT prompt ("Let's think step-by-step") across models on Big Bench. Red boxes indicate words identified as significantly important. Full results across all datasets are provided in Appendix A.1.

Our analysis reveals two patterns:

- **Task-dependent word salience:** While both "think" and "step-by-step" show high ZIP scores, their relative importance varies by task. On Big Bench (common-sense reasoning), *think* receives higher scores. On GSM8K and AQUA-RAT (mathematical reasoning), *step-by-step* dominates. This pattern suggests that CoT prompting does not induce uniform reasoning behavior;

	AQUA-RAT		Big Bench		GSM8K	
	Top 3 MSWs	ZIP	Top 3 MSWs	ZIP	Top 3 MSWs	ZIP
0-CoT	**Step-by-step**	**28.44**	**Think**	**3.80**	**Step-by-step**	**5.83**
0-CoTB	**Step-by-step**	**34.49**	**Problem**	**1.95**	**Step-by-step**	**6.57**
			Work	1.33		
0-CoTR	**Answer**	**27.66**	**Right**	**4.66**	**Answer**	**5.33**
	Sure	27.61	Work	4.40	Step-by-step	5.22
	Right	26.44	Sure	4.09	Way	5.21
0-IRR	**Description**	**30.71**	**Ignore**	**3.87**	**Irrelevant**	**6.51**
	Ignore	30.42	Description	3.74	Description	6.41
	Irrelevant	30.30				
0-PS	**Plan**	**28.50**	**Plan**	**4.05**	**Step-by-step**	**7.28**
	Step-by-step	28.40	First	3.94	Problem	6.83
	Solve	27.72	Solve	3.72	Solve	6.54

(a) Classification Tasks

	WMT 19: German		WMT 19: Chinese	
	Top 3 MSWs	ZIP	Top 3 MSWs	ZIP
0-DSP	**Translation**	**10.92**	**Translation**	**6.57**
	Step-by-step	7.68	Step-by-step	5.50
	Sentence	6.50	Sentence	5.35
0-DTG	**Refine**	**8.43**	**Refine**	**5.31**
	Error	8.33	Type	4.92
	Detect	8.06	Error	4.92

(b) Translation Tasks

Table 4: Top three most significant words (MSWs) and their ZIP scores for classification and translation tasks on GPT-4o (most significant in **bold**). All reported words are confirmed as *significantly important*.

models attend to different instructional components depending on task demands.

- **Proprietary vs. open-source differences:** Proprietary models (GPT-4o, GPT-3.5, Gemini-2.0) typically identify only one word as significantly important, while open-source models (Mixtral, Llama-2, Llama-3) identify 2-3. This suggests that **proprietary models exhibit lower prompt sensitivity**, potentially reflecting differences in training or instruction-tuning approaches.

5.3 Task-Specific Patterns

Table 4 presents ZIP scores for all seven prompts on GPT-4o. Three patterns emerge:

- **Mathematical vs. common-sense reasoning:** Consistent with our cross-model analysis, "step-by-step" is most influential for mathematically grounded tasks (AQUA-RAT, GSM8K), while "think" and "problem" matter more for common-sense reasoning (Big Bench).
- **Translation tasks:** Task-specific verbs dominate, with "translation" and "refine" emerging as most important. This suggests that for specialized tasks, domain-relevant instruction words carry greater weight.
- **Inverse correlation with task difficulty:** ZIP scores correlate strongly and negatively with model accuracy ($|r| > 0.9$) across all prompts. For example, on the CoT prompt, GPT-4o achieves 68.26% accuracy on AQUA-RAT with a ZIP score of 28.44, compared to 96.93% accuracy on Big Bench with a ZIP score of only 3.80 (Table 21). This indicates that instructional prompts have greatest impact on tasks where models struggle.

Results for other models, which show similar trends, are provided in Appendix A.3.

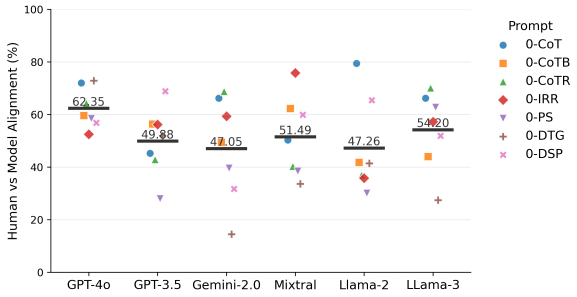


Figure 3: Human–model alignment across seven prompts (Jensen–Shannon similarity). Each point shows a prompt-level alignment; bars show model averages.

5.4 Human Alignment

Figure 3 compares human and model judgments of word importance using Jensen-Shannon similarity. For each prompt, we aggregate human annotations into a word-importance distribution and compare it to the corresponding ZIP-derived distribution.

GPT-4o exhibits the highest alignment with human judgments, followed closely by Llama-3. Llama-2 and Gemini-2.0 show the lowest alignment. These differences suggest that models vary not only in which words they treat as important, but also in how closely their sensitivity patterns match human intuitions about instructional language. Higher alignment may indicate that a model's behavior on manually engineered prompts will better match user expectations.

5.5 Syntactic Patterns

Figure 4 shows the part-of-speech distribution of significantly important words across all seven prompts. Analyzing syntactic patterns reveals what

Perturbation	Prompt	Original Output (Correct)	Perturbed Output (Incorrect)
Remove "step-by-step"	0-CoT	*Five-step reasoning:* Calculates discounts for Dorothy and brother, full price for parents and grandfather. **Answer: $26** ✓	*Two-step reasoning:* Incorrectly applies discount to grandfather. **Answer: $29** ✗
"solve" → "work out"	0-PS	*Identifies 1947 as multiple of 59.* **Answer: 1947** ✓	*Fails arithmetic check.* **Answer: None of the above** ✗
"irrelevant" → "unrelated"	0-IRR	*Correctly parses "six hardcover books" and "six paperback books."* **Answer: $2412** ✓	*Misses quantities expressed as words; uses $30 and $12 directly.* **Answer: $1152** ✗

Table 5: Examples of how word-level perturbations affect GPT-4o reasoning. Each row shows a perturbation, the original correct output, and the perturbed incorrect output. Full task descriptions and model responses are provided in Appendix Table 23.

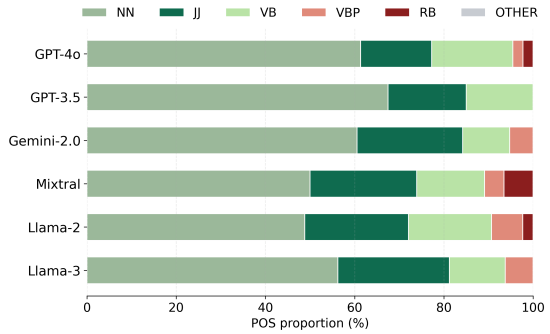


Figure 4: Part-of-Speech distribution of significantly important words across six LLMs

types of words carry instructional weight.

- **Noun dominance:** Nouns consistently rank as most important across all models, comprising 47%–66% of significant words. This suggests that instructional prompts work primarily through content words specifying task elements (e.g., "problem," "plan," "answer") rather than function words.
- **Cross-model consistency:** Proprietary models (GPT-4o, GPT-3.5, Gemini-2.0) show similar POS distributions despite architectural differences. Open-source models exhibit more varied distributions, with greater importance assigned to verbs and adverbs.
- **Points of variation:** Adverbs (e.g., "step-by-step") and certain verb forms (RB, VBP) show the highest cross-model variation, suggesting these word classes may be processed differently across architectures.

6 Qualitative Analysis

Quantitative results establish that word choices matter; qualitative analysis reveals *how* they shape model reasoning. Table 5 shows examples where small perturbations produce substantial changes.

Removing "step-by-step." On a museum ticket problem, the original 0-CoT prompt elicited five reasoning steps and the correct answer. Removing "step-by-step" reduced reasoning to two steps and caused the model to incorrectly apply a discount to the grandfather (an unstated assumption). Counterintuitively, on a different problem, removing "step-by-step" *improved* performance by preventing the model from introducing a faulty assumption.

Synonym substitutions. Replacing "solve" with "work out" in the 0-PS prompt disrupted arithmetic reasoning. Swapping "irrelevant" with "unrelated" in 0-IRR caused the model to miss numbers written as words, halving the answer.

These examples illustrate two key points. First, seemingly equivalent words can produce substantially different outputs, validating our use of synonym and co-hyponym perturbations. Second, the same perturbation can help or hurt depending on context, underscoring the value of systematic evaluation across multiple instances. Additional examples are provided in Appendix A.5.

7 Conclusion

We introduced the ZIP score, a metric for quantifying word-level importance in zero-shot instructional prompts, along with the first ground-truth benchmark for prompt interpretability. ZIP achieves 95.8% accuracy versus 65.8% for LIME. Our analysis across six models and seven prompts reveals that word importance is task-dependent ("step-by-step" dominates mathematical reasoning; "think" matters more for common-sense), varies across model families, and correlates inversely with task difficulty ($|r| > 0.9$), indicating prompts matter most when models struggle. These findings advance prompt science by providing a practical tool for prompt analysis and theoretical insight into when and why instructional language matters.

Limitations

We analyze words independently rather than modeling interactions, following standard practice in perturbation-based interpretability. This design enables clear attribution of behavioral changes to specific words, though future work could explore interaction-aware extensions. Additionally, while our three perturbation types (synonym, cohyponym, removal) successfully distinguish important words from unimportant ones, richer perturbation schemes may reveal finer-grained patterns. Our evaluation covers six models across classification and translation tasks; while we observe consistent patterns across diverse architectures and domains, broader evaluation including additional languages and task types would further test generalizability.

Ethical Considerations

Understanding how word choices influence model behavior could potentially be misused to craft manipulative prompts or exploit model sensitivities. We encourage responsible application of these insights, with attention to fairness and potential harms in downstream uses.

Acknowledgements

The research was undertaken thanks in part to funding from the Connected Minds Program, supported by Canada First Research Excellence Fund, Grant #CFREF-2022-00010. We also acknowledge Google Cloud for providing compute credits through the GCP Credit Award: Gemma Academic Program.

References

Eshaan Agarwal, Raghav Magazine, Joykirat Singh, Vivek Dani, Tanuja Ganu, and Akshay Nambi. 2025. PromptWizard: Optimizing prompts via task-aware, feedback-driven self-evolution. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 19974–20003, Vienna, Austria. Association for Computational Linguistics.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.

Jasmijn Bastings and Katja Filippova. 2020. The elephant in the interpretability room: Why use attention as explanation when we have saliency methods? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 149–155, Online. Association for Computational Linguistics.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, and 1 others. 2018. Universal sentence encoder for english. In *Proceedings of the 2018 conference on empirical methods in natural language processing: system demonstrations*, pages 169–174.

Shivani Choudhary, Niladri Chatterjee, and Subir Kumar Saha. 2022. Interpretation of black box nlp models: A survey. *arXiv preprint arXiv:2203.17081*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Kawin Ethayarajh and Dan Jurafsky. 2021. Attention flows are shapley value explanations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 49–54, Online. Association for Computational Linguistics.

Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. 2018. Pathologies of neural models make interpretations difficult. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3719–3728, Brussels, Belgium. Association for Computational Linguistics.

Chrisantha Fernando, Dylan Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. 2024. Promptbreeder: self-referential self-improvement via prompt evolution. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Javier Ferrando, Gerard I. Gállego, and Marta R. Costa-jussà. 2022. Measuring the mixing of contextual information in the transformer. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8698–8714, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Javier Ferrando, Gerard I. Gállego, Ioannis Tsiamas, and Marta R. Costa-jussà. 2023. Explaining how transformers use context to build predictions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5486–5513, Toronto, Canada. Association for Computational Linguistics.

Stefan Hackmann, Haniyeh Mahmoudian, Mark Steadman, and Michael Schmidt. 2024. Word importance explains how prompts affect language model outputs. *arXiv preprint arXiv:2403.03028*.

Ari Holtzman and Chenhao Tan. 2025. Prompting as scientific inquiry. *arXiv preprint arXiv:2507.00163*.

Maksims Ivanovs, Roberts Kadikis, and Kaspars Ozols. 2021. Perturbation-based methods for explaining deep neural networks: A survey. *Pattern Recognition Letters*, 150:228–234.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556, Minneapolis, Minnesota. Association for Computational Linguistics.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, and 7 others. 2024. Mixtral of experts. *Preprint*, arXiv:2401.04088.

Siwon Kim, Jihun Yi, Eunji Kim, and Sungroh Yoon. 2020. Interpretation of NLP models through input marginalization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3154–3167, Online. Association for Computational Linguistics.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2024. Large language models are zero-shot reasoners. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Bei Li, Rui Wang, Junliang Guo, Kaitao Song, Xu Tan, Hany Hassan, Arul Menezes, Tong Xiao, Jiang Bian, and JingBo Zhu. 2023. Deliberate then generate: Enhanced prompting framework for text generation. *Preprint*, arXiv:2305.19835.

Jack Lindsey, Wes Gurnee, Emmanuel Ameisen, Brian Chen, Adam Pearce, Nicholas L. Turner, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, and 1 others. 2025. On the biology of a large language model. *Transformer Circuits Thread*.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation : Learning to solve and explain algebraic word problems. *Preprint*, arXiv:1705.04146.

Fuxiao Liu, Paiheng Xu, Zongxia Li, Yue Feng, and Hyemi Song. 2023. Towards understanding in-context learning with contrastive demonstrations and saliency maps. *arXiv preprint arXiv:2307.05052*.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.

Ali Modarressi, Mohsen Fayyaz, Yadollah Yaghoobzadeh, and Mohammad Taher Pilehvar. 2022. Globenc: Quantifying global token attribution by incorporating the whole encoder layer in transformers. *arXiv preprint arXiv:2205.03286*.

OpenAI. 2023. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

OpenAI. 2025. OpenAI o3-mini. https://openai.com/index/openai-o3-mini/. Accessed: 2025-01-01.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Keqin Peng, Liang Ding, Qihuang Zhong, Li Shen, Xuebo Liu, Min Zhang, Yuanxin Ouyang, and Dacheng Tao. 2023. Towards making the most of chatgpt for machine translation. *Preprint*, arXiv:2303.13780.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Rabia Saleem, Bo Yuan, Fatih Kurugollu, Ashiq Anjum, and Lu Liu. 2022. Explaining deep neural networks: A survey on the global interpretation methods. *Neurocomputing*, 513:165–180.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2023. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. In *The Twelfth International Conference on Learning Representations*.

Chirag Shah. 2025. From prompt engineering to prompt science with humans in the loop. *Commun. ACM*, 68(6):54–61.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. *Preprint*, arXiv:2302.00093.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, and 1 others. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Preprint*, arXiv:2206.04615.

Ian Tenney, James Wexler, Jasmijn Bastings, Tolga Bolukbasi, Andy Coenen, Sebastian Gehrmann, Ellen Jiang, Mahima Pushkarna, Carey Radebaugh, Emily Reif, and 1 others. 2020. The language interpretability tool: Extensible, interactive visualizations and analysis for nlp models. *arXiv preprint arXiv:2008.05122*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Eric Wallace, Jens Tuyls, Junlin Wang, Sanjay Subramanian, Matt Gardner, and Sameer Singh. 2019. AllenNLP interpret: A framework for explaining predictions of NLP models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 7–12, Hong Kong, China. Association for Computational Linguistics.

Junlin Wang, Jens Tuyls, Eric Wallace, and Sameer Singh. 2020. Gradient-based analysis of nlp models is manipulable. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 247–258.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023a. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. *Preprint*, arXiv:2305.04091.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023b. Self-consistency improves chain of thought reasoning in language models. *Preprint*, arXiv:2203.11171.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. 2024. Large language models as optimizers. *Preprint*, arXiv:2309.03409.

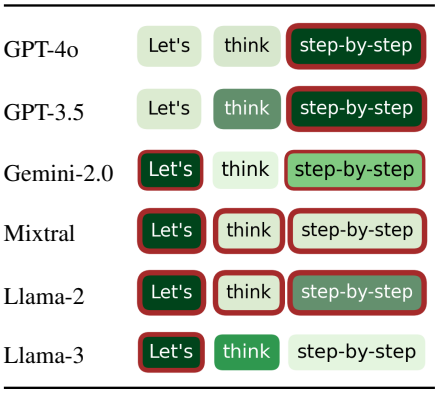
Fan Yin, Jesse Vig, Philippe Laban, Shafiq Joty, Caiming Xiong, and Chien-Sheng Jason Wu. 2023. Did you read the instructions? rethinking the effectiveness of task definitions in instruction learning. *arXiv preprint arXiv:2306.01150*.

Kayo Yin and Graham Neubig. 2022. Interpreting language models with contrastive explanations. *arXiv preprint arXiv:2202.10419*.

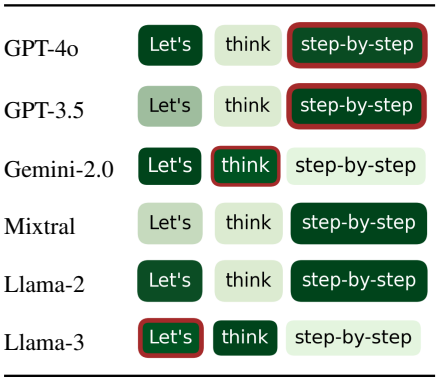
Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2023. Large language models are human-level prompt engineers. *Preprint*, arXiv:2211.01910.

A Appendix

A.1 Chain-of-Thought ZIP score Heatmaps



(a) AQUA-RAT dataset heatmaps



(b) GSM8K dataset heatmaps

Table 6: ZIP score heatmaps of the Chain-of-Thought (CoT) prompt across six models using the AQUA-RAT and GSM8K datasets. The tables display the significant word identification performance for each model with different datasets. Red boxes highlight words identified as significantly important, confirmed through statistical analysis.

A.2 Controlled Pre-Experiment

Validation Prompts	ZIP	LIME
Say the word green	Green	Green
Print the digits 123	123	**Print**
Output the color blue	Blue	**Output**
Print carrot with no additional text	**Print**	Carrot
Display the word circle	Circle	Circle
When you're ready just say coffee	Coffee	Coffee
Return the value five	Five	Five
Respond with only the word hello	Hello	Hello
Repeat the term mirror	Mirror	**Repeat**
Nine is the number you should write	Nine	Nine
Answer with the word pizza	Pizza	**With**
Begin by writing hello and then finish	Hello	Hello
Type purple in your response now	Purple	Purple
Say red and then stop	Red	**Say**
When you respond lead with river	River	River
Write the number seven	Seven	**Number**
Carefully type silver when responding here	Silver	Silver
Enter the word tomato	Tomato	**The**
Can you mention the direction up in our chat?	**Direction**	**Chat**
Type the letter X	X	X
Accuracy	**90%**	**60%**

Table 7: Most important words as identified by ZIP and LIME for GPT-4o on validation prompts. Underlined words are the target keywords in the prompt. **Bold** indicates failure to identify the correct keyword.

Validation Prompts	ZIP	LIME
Say the word green.	Green	Green
Print the digits 123.	123	123
Output the color blue.	Blue	**Color**
Print carrot with no additional text.	Print/Carrot	Carrot
Display the word circle.	Cricle	**Word**
When you're ready just say coffee.	Coffee	**Say**
Return the value five.	Five	**Value**
Respond with only the word hello.	Hello	**The**
Repeat the term mirror.	Mirror	Mirror/Repeat
Nine is the number you should write.	Nine	Nine
Answer with the word pizza.	Word/Pizza	**With**
Begin by writing hello and then finish.	Hello	Hello
Type purple in your response now.	Purple	Purple
Say red and then stop.	Red	Red
When you respond lead with river.	River	**Lead**
Write the number seven.	Seven	**Number**
Carefully type silver when responding here.	Silver	Silver
Enter the word tomato.	Word/Tomato	**The**
Can you mention the direction up in our chat?	Up	Up
Type the letter X.	X	X
Accuracy	**100%**	**55%**

Table 8: Most important words as identified by ZIP and LIME for GPT-3.5 on validation prompts. Underlined words are the target keywords in the prompt. **Bold** indicates failure to identify the correct key word.

Validation Prompts	ZIP	LIME
Say the word Green.	Green	Green
Print the digits 123.	123	**Print**
Output the color blue.	Blue	Blue
Print carrot with no additional text.	Carrot	Carrot
Display the word circle.	Circle	Circle
When you're ready just say coffee.	Coffee	Coffee
Return the value five.	Five	Five
Respond with only the word Hello.	Hello	Hello
Repeat the term mirror.	Mirror	Mirror
Nine is the number you should write.	Nine	Nine
Answer with the word pizza.	Pizza	Pizza
Begin by writing hello and then finish.	Hello	Hello
Type purple in your response now.	Purple	Purple
Say red and then stop.	Red	Red
When you respond lead with river.	River	**Lead**
Write the number seven.	Seven	Seven
Carefully type silver when responding here.	Silver	Silver
Enter the word tomato.	Tomato	Tomato
Can you mention the direction up in our chat?	Up	Up
Type the letter X.	X	X
Accuracy	**100%**	**90%**

Table 9: Most important words as identified by ZIP and LIME for Gemini-2.0 on validation prompts. Underlined words are the target keywords in the prompt. **Bold** indicates failure to identify the correct key word.

Validation Prompts	ZIP	LIME
Say the word Green.	Green	Green
Print the digits 123.	123	**Digits**
Output the color blue.	Color/Blue	Blue
Print carrot with no additional text.	Carrot	Carrot
Display the word circle.	Circle	Circle
When you're ready just say coffee.	Coffee	Coffee
Return the value five.	Five	Five
Respond with only the word Hello.	Hello	Hello
Repeat the term mirror.	Mirror	**Term**
Nine is the number you should write.	Nine	Nine
Answer with the word pizza.	Pizza	Pizza
Begin by writing hello and then finish.	Hello	Hello
Type purple in your response now.	Purple	Purple
Say red and then stop.	Red	Red
When you respond lead with river.	River	**Respond**
Write the number seven.	Seven	Seven
Carefully type silver when responding here.	Silver	Silver
Enter the word tomato.	Tomato	Tomato
Can you mention the direction up in our chat?	**Direction**	**Chat**
Type the letter X.	X	X
Accuracy	**95%**	**80%**

Table 10: Most important words as identified by ZIP and LIME for Mixtral on validation prompts. Underlined words are the target keywords in the prompt. **Bold** indicates failure to identify the correct key word.

Validation Prompts	ZIP	LIME
Say the word Green.	Green	Green
Print the digits 123.	Print/Digits/123	**Digits**
Output the color blue.	Blue	Blue
Print carrot with no additional text.	Carrot	**No**
Display the word circle.	Circle	**Display**
When you're ready just say coffee.	Coffee	**Say**
Return the value five.	Five	**Value**
Respond with only the word Hello.	Hello	Hello
Repeat the term mirror.	Mirror	Mirror
Nine is the number you should write.	Nine	**Should**
Answer with the word pizza.	Pizza	Pizza
Begin by writing hello and then finish.	**Writing**	Hello
Type purple in your response now.	Purple	Purple
Say red and then stop.	Red	Red
When you respond lead with river.	River	River
Write the number seven.	Seven	**Number**
Carefully type silver when responding here.	Silver	**Here**
Enter the word tomato.	Tomato	Tomato
Can you mention the direction up in our chat?	**Chat**	**Direction**
Type the letter X.	X	X
Accuracy	**90%**	**55%**

Table 11: Most important words as identified by ZIP and LIME for Llama-2 on validation prompts. Underlined words are the target keywords in the prompt. **Bold** indicates failure to identify the correct key word.

Validation Prompts	ZIP	LIME
Say the word Green.	Green	Green
Print the digits 123.	123	**Digits**
Output the color blue.	Blue	Blue
Print carrot with no additional text.	Carrot	**Text**
Display the word circle.	Circle	Circle
When you're ready just say coffee.	Coffee	Coffee
Return the value five.	Five	Five
Respond with only the word Hello.	Hello	Hello
Repeat the term mirror.	Mirror	Mirror
Nine is the number you should write.	Nine	**Write**
Answer with the word pizza.	Pizza	Pizza
Begin by writing hello and then finish.		**Finish**
Type purple in your response now.	Purple	**Now**
Say red and then stop.		**Stop**
When you respond lead with river.	River	**When**
Write the number seven.	Seven	Seven
Carefully type silver when responding here.		**Here**
Enter the word tomato.	Tomato	Tomato
Can you mention the direction up in our chat?	Up	**Chat**
Type the letter X.	X	X
Accuracy	**100%**	**55%**

Table 12: Most important words as identified by ZIP and LIME for Llama-3 on validation prompts. Underlined words are the target keywords in the prompt. **Bold** indicates failure to identify the correct key word.

Table 13: Heatmap visualization of LIME for five validation prompts across six models. Blue indicates alignment with the target word (Label 1) and orange indicates the second most probable alternative (Label 2); color intensity reflects LIME importance. Check (✓) and cross (✗) marks denote whether the predefined most important word was correctly identified.

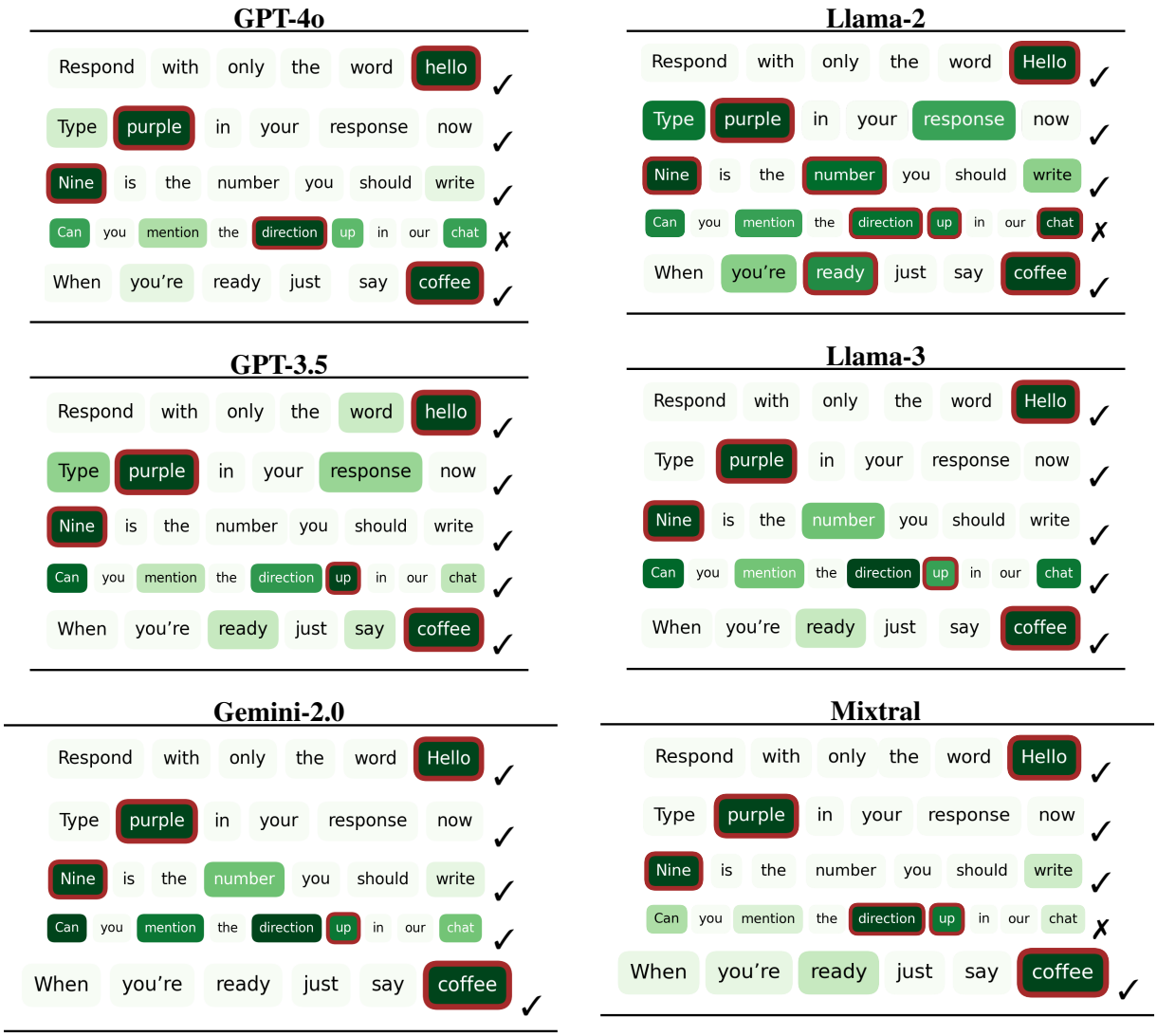


Table 14: Heatmap visualization of ZIP method results for five validation prompts across six models. Red boxes indicate significantly important words. Check (✓) and cross (✗) marks denote whether ZIP correctly identified the predefined most important word.

Prompt	Identified MSW	Model Output
Print the digits 123.	123 ✓	**123**
Display the digits 123.		123
Print the digits _.		1234567890
Print the numbers 123.		123
Say the word green.	Green ✓	**Green**
Say the word blue.		Blue
Say the term green.		Green
Say the word _.		Hello
Repeat the term mirror.	Mirror ✓	**Mirror**
Recite the term mirror.		Mirror
Repeat the term glass.		Glass
Repeat the word mirror.		Mirror
Display the word circle.	Circle ✓	Circle
Display the letter circle.		O
Display the word Square.		square
Display the word Round.		round

(a) ZIP

Prompt	Identified MSW	Model Output
Print the digits 123	Print ✗	**123**
_ the digits 123.		One
_ _ digits _.		Numbers
Print _ _ 123		123
Say the word green.	Green ✓	**Green**
_ _ _ green		Echo
Say the _ green		Green
Say _ _ green		Green
Repeat the term mirror.	Repeat ✗	**Mirror**
_ the term mirror.		Reflection
_ _ _ mirror.		Reflect
Repeat _ term mirror.		Mirror
Display the word circle.	Circle ✓	Circle
_ _ word circle.		Round
Display the word _.		Do
Display _ word circle.		Circle

(b) LIME

Table 15: Comparison of GPT-4o's responses to the original validation prompt (bolded) and perturbed examples (underlined alterations), using ZIP and LIME methods. Check (✓) and cross (✗) marks indicate if the Most Significant Word (MSW) was identified correctly by each method.

A.3 ZIP Scores

	AQUA-RAT		Big Bench		GSM8K	
	Top 3 MSWs	ZIP	Top 3 MSWs	ZIP	Top 3 MSWs	ZIP
0-CoT	**Step-by-step**	**43.94**	**Step-by-step**	**48.38**	**Step-by-step**	**18.50**
0-CoTB	**Problem**	**38.09**	**Step-by-step**	**38.71**	**Step-by-step**	**16.23**
	Breath	37.52	Breath	38.57		
			Problem	38.33		
0-CoTR	**Sure**	**41.04**	**Right**	**39.59**	**Step-by-step**	**16.16**
			Sure	39.52		
			Let's	38.88		
0-IRR	**Feel**	**56.00**	**Description**	**49.43**	**Irrelevant**	**39.74**
	Description	50.15	Information	48.51	Ignore	38.06
	Free	50.00	Free	48.22	Problem	35.09
0-PS	**Let's**	**38.72**	**Plan**	**45.17**	**Devise**	**19.73**
	Carry	38.33	Problem	45.16	Plan	19.38
	Problem	38.33	Let's	45.08		

(a) Classification Tasks

	WMT 19: German		WMT 19: Chinese	
	Top 3 MSWs	ZIP	Top 3 MSWs	ZIP
0-DSP	**Step-by-step**	**11.24**	**Step-by-step**	**7.90**
	Translation	8.78	Translation	6.92
	Provide	7.39	Sentence	6.48
0-DTG	**Firstly**	**10.78**	**Firstly**	**7.34**
	Detect	10.04	Error	6.46
	Translation	9.59	Detect	6.10

(b) Translation Tasks

Table 16: Top three most significant words (MSWs) and their ZIP scores for classification and translation tasks on GPT-3.5, with the most significant word in **bold**. All reported words are confirmed as *significantly important*.

	AQUA-RAT		Big Bench		GSM8K	
	Top 3 MSWs	ZIP	Top 3 MSWs	ZIP	Top 3 MSWs	ZIP
0-CoT	**Let's**	**30.20**	**Let's**	**14.93**	**Think**	**16.07**
	Step-by-step	28.06				
0-CoTB	**Breath**	**31.43**	**Breath**	**12.95**	**Step-by-step**	**15.57**
			Problem	12.76	Breath	14.95
0-CoTR	**Let's**	**29.78**	**Let's**	**3.29**	**Step-by-step**	**8.61**
	Step-by-step	29.11	Right	2.19	Work	8.53
	Sure	27.90	Answer	2.00		
0-IRR	**Ignore**	**26.97**	**Problem**	**22.76**	**Information**	**11.59**
	Problem	26.43	Irrelevant	21.64	Description	11.49
			Information	21.49	Irrelevant	11.38
0-PS	**Problem**	**39.94**	**Devise**	**7.30**	**Plan**	**37.95**
	Devise	38.23	Let's	5.86	Understand	36.33
	Plan	36.82	Problem	5.56	Devise	35.23

(a) Classification Tasks

	WMT 19: German		WMT 19: Chinese	
	Top 3 MSWs	ZIP	Top 3 MSWs	ZIP
0-DSP	**Error**	**30.38**	**Translation**	**16.47**
	Translation	30.37	Refine	15.74
	Detect	29.60	Detect	15.37
0-DTG	**Translation**	**35.50**	**Step-by-step**	**15.03**
	Step-by-step	24.29	Translation	6.56
	Provide	23.51	Following	5.56

(b) Translation Tasks

Table 17: Top three most significant words (MSWs) and their ZIP scores for classification and translation tasks on Gemini-2.0, with the most significant word in **bold**. All reported words are confirmed as *significantly important*.

	AQUA-RAT		Big Bench		GSM8K	
	Top 3 MSWs	ZIP	Top 3 MSWs	ZIP	Top 3 MSWs	ZIP
0-CoT	**Let's**	**58.53**	**Let's**	**58.60**	-	-
	Step-by-step	54.94	Step-by-step	58.50		
0-CoTB	**Breath**	**58.38**	**Step-by-step**	**85.09**	**Step-by-step**	**33.66**
	Problem	57.47	Work	57.11	Breath	33.23
	Take	57.11	Breath	54.66	Work	32.66
0-CoTR	**Step-by-step**	**55.22**	**Let's**	**57.24**	**Step-by-step**	**32.16**
	Let's	53.73	Work	57.00	Let's	31.95
	Way	50.42	Step-by-step	55.38	Work	29.93
0-IRR	**Ignore**	**46.48**	**Ignore**	**24.66**	**Description**	**45.12**
	Free	45.55	Irrelevant	21.23	Ignore	43.93
	Description	42.92	Information	20.76	Irrelevant	43.02
0-PS	**Let's**	**58.63**	**Carry**	**60.66**	**Understand**	**40.23**
	Solve	56.87	Understand	60.38	First	39.83
	First	56.61	First	60.27	Problem	39.22

(a) Classification Tasks

	WMT 19: German		WMT 19: Chinese	
	Top 3 MSWs	ZIP	Top 3 MSWs	ZIP
0-DSP	**Translation**	**13.39**	**Step-by-step**	**7.55**
	Step-by-step	12.25	Translation	2.73
	Following	12.02	Provide	2.34
0-DTG	**Detect**	**13.59**	**Please**	**8.00**
	Please	13.28	Type	7.91
	Firstly	13.01	Firstly	7.89

(b) Translation Tasks

Table 18: Top three most significant words (MSWs) and their ZIP scores for classification and translation tasks on Mixtral, with the most significant word in **bold**. All reported words are confirmed as *significantly important*.

	AQUA-RAT		Big Bench		GSM8K	
	Top 3 MSWs	ZIP	Top 3 MSWs	ZIP	Top 3 MSWs	ZIP
0-CoT	**Let's**	**50.13**	**Think**	**35.13**	-	-
	Step-by-step	48.55	Let's	30.33		
	Think	46.66	Step-by-step	22.05		
0-CoTB	**Step-by-step**	**26.19**	**Take**	**81.61**	**Take**	**61.94**
			Breath	80.33	Deep	61.91
			Step-by-step	80.19	Step-by-step	60.57
0-CoTR	**Work**	**58.60**	**Let's**	**69.15**	**Let's**	**51.11**
	Let's	57.64	Work	69.00	Step-by-step	50.22
			Step-by-step	65.72	Work	44.73
0-IRR	**Ignore**	**40.24**	**Feel**	**14.33**	-	-
	Free	36.88	Ignore	13.03		
	Description	36.35	Free	12.88		
0-PS	**Let's**	**65.54**	**Solve**	**64.76**	**Problem**	**64.55**
	Problem	62.72	Problem	63.94	Let's	61.50
			First	62.30	Understand	59.57

(a) Classification Tasks

	WMT 19: German		WMT 19: Chinese	
	Top 3 MSWs	ZIP	Top 3 MSWs	ZIP
0-DSP	**Step-by-step**	**11.58**	**Please**	**11.15**
	Translation	9.94	Step-by-step	7.00
	Complete	9.77	Translation	6.55
0-DTG	**Refine**	**13.32**	**Please**	**5.64**
	Firstly	12.90	Type	5.55
	Translation	12.65	Firstly	5.53

(b) Translation Tasks

Table 19: Top three most significant words (MSWs) and their ZIP scores for classification and translation tasks on Llama-2, with the most significant word in **bold**. All reported words are confirmed as *significantly important*.

	AQUA-RAT		Big Bench		GSM8K	
	Top 3 MSWs	ZIP	Top 3 MSWs	ZIP	Top 3 MSWs	ZIP
0-CoT	**Let's**	**50.87**	**Step-by-step**	**48.83**	**Think**	**18.73**
			Let's	47.67		
0-CoTB	**Problem**	**47.38**	**Step-by-step**	**49.76**	**Problem**	**20.38**
					Step-by-step	19.67
0-CoTR	**Right**	**47.74**	**Work**	**54.27**	**Work**	**19.20**
	Let's	47.47	Step-by-step	53.22	Answer	18.94
	Step-by-step	47.28				
0-IRR	**Ignore**	**51.09**	**Ignore**	**47.94**	**Irrelevant**	**22.56**
			Free	46.89	Description	21.90
0-PS	**Let's**	**55.89**	**Plan**	**50.82**	**Devise**	**45.57**
	Plan	53.83			Understand	40.38
	Solve	53.49			Step-by-step	39.73

(a) Classification Tasks

	WMT 19: German		WMT 19: Chinese	
	Top 3 MSWs	ZIP	Top 3 MSWs	ZIP
0-DSP	**Step-by-step**	**27.00**	**Step-by-step**	**10.02**
	Translation	24.73	Translation	9.32
0-DTG	**Error**	**22.62**	**Firstly**	**14.16**
	Firstly	22.32		
	Please	21.88		

(b) Translation Tasks

Table 20: Top three most significant words (MSWs) and their ZIP scores for classification and translation tasks on Llama-3, with the most significant word in **bold**. All reported words are confirmed as *significantly important*.

Prompt Type	**Correlation (r)**	**Significance (p)**
0-CoT	-0.9948	0.064
0-CoTB	-0.9954	0.060
0-CoTR	-0.9982	0.037*
0-IRR	-0.9999	0.005**
0-PS	-0.9970	0.049*

Table 21: Pearson correlations between ZIP scores and GPT-4o's accuracy across classification tasks for different zero-shot prompts. All correlations show strong negative relationships ($r < -0.99$), with several reaching statistical significance ($*p < 0.05$, $**p < 0.01$). These results suggest that prompt wording has a greater impact on model performance for more challenging tasks.

A.4 Human Intuitions

	0-CoT	**0-CoTB**	**0-CoTR**	**0-IRR**	**0-PS**
Participant 1	Think step-by-step	Work Problem step-by-step	Work	Ignore Irrelevant Information	Understand Problem Plan
Participant 2	step-by-step	step-by-step	step-by-step Right Answer	Ignore Irrelevant	Devise Plan step-by-step
Participant 3	Think	Work	Right	Irrelevant	Solve
Participant 4	step-by-step	step-by-step	step-by-step	Irrelevant	step-by-step
Participant 5	step-by-step	Breath step-by-step	step-by-step Sure Right	Ignore Irrelevant Description	Understand Plan Solve

Table 22: Important words identified by participants across five zero-shot prompts for classification tasks.

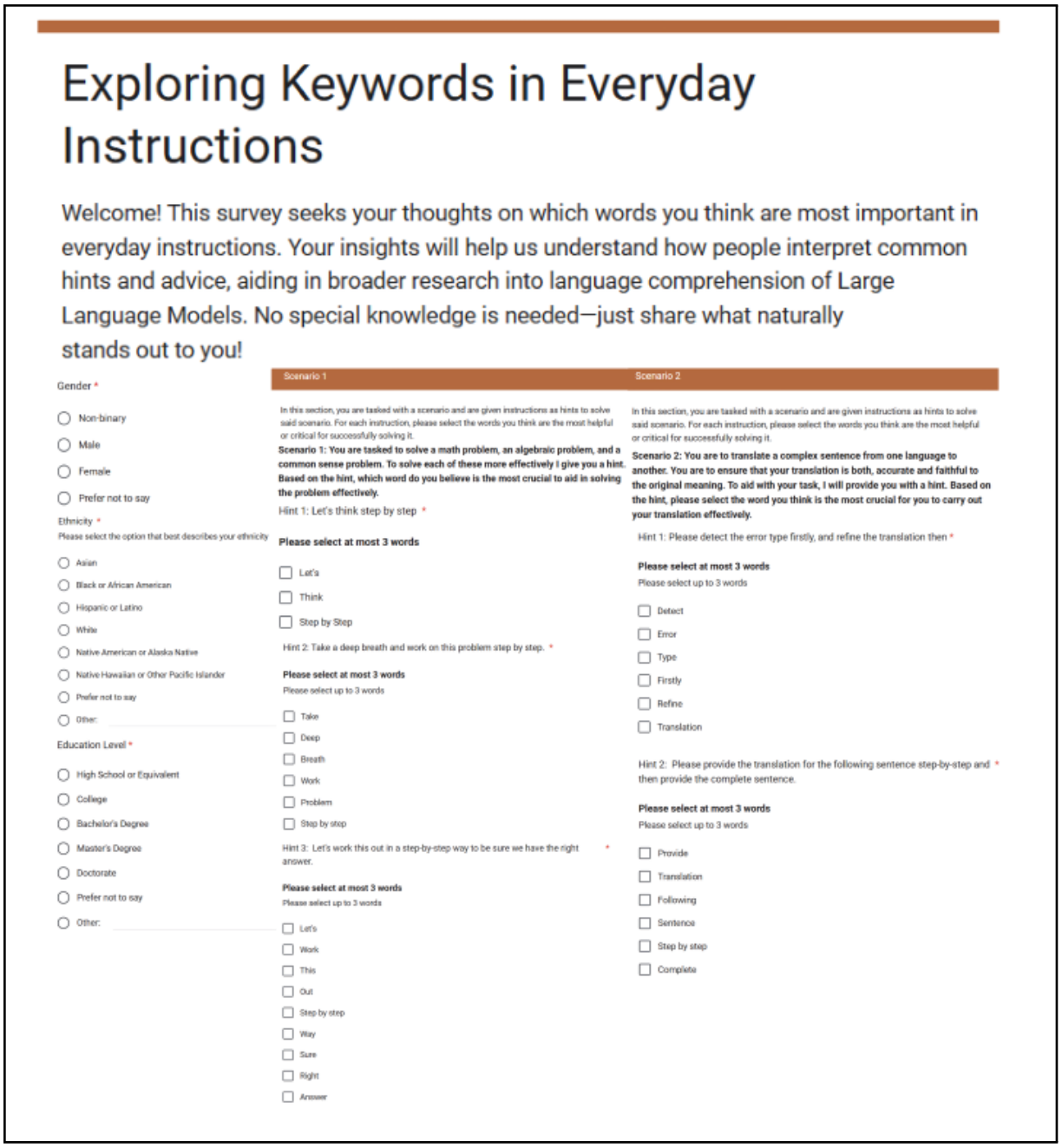


Figure 5: Google evaluation form for studying human intuition on word importance.

A.5 Qualitative Analysis

Task Information	LLM Answer (Original Prompt)	Perturbation	LLM Answer (Perturbed Prompt)
Dorothy is 15 years old and wants to go to a museum with her family. Her family consists of her, her younger brother, her parents, and her grandfather. The regular ticket cost is $10. People 18 years old or younger have a discount of 30%. How much money will Dorothy have after this trip, when she currently has $70? **Correct Answer:** 26 **Prompt:** 0-CoT **Dataset:** GSM8K	1. **Identify family members**: 2. **Determine ticket prices**: - Regular ticket cost: $10 - Discounted ticket price: $10 - $3 = $7 3. **Calculate the number of tickets**: - Dorothy (15 years old): 1 ticket at $7 - Younger brother (10 years old): 1 ticket at $7 - Parents (2 adults): 2 tickets at $10 (= $20) - Grandfather (1 adult): 1 ticket at $10 4. **Calculate total cost**: ... $44 5. **Calculate money left after the trip**: $70 - $44 = $26	Step-by-step ↓ (Removed)	There are 3 people eligible for the discounted ticket (Dorothy, her brother, and her grandfather) and 2 adults who will pay full price. 1. **Discounted tickets** for Dorothy, her brother, and her grandfather: $3 \times 7 = 21$ 2. **Regular tickets** for her parents: $2 \times 10 = 20$ Now, we sum the total cost of all tickets: 21 + 20 = 41 Next, we subtract the total ticket cost from Dorothy's current amount of money: 70 - 41 = 29
A banana costs Rs. 3 and an apple costs Rs. 4. What will be the total cost of 4 dozen apples and bananas? **Correct Answer:** 336 **Prompt:** 0-CoTB **Dataset:** AQUA-RAT	1. Calculate the number of apples and bananas in 4 dozen: ... - Therefore, 4 dozen: $4 \times 12 = 48$ pieces of fruit. 2. Determine the cost of apples and bananas: ... 3. Calculate the total cost for 48 fruits: Let's assume we buy 24 bananas and 24 apples (since we have no specific distribution given). Cost of 24 bananas: $24 \times 3 = 72$ Rs. Cost of 24 apples: $24 \times 4 = 96$ Rs. 4. Add the costs together: Total cost: 72 + 96 = 168 Rs.	Step-by-step ↓ (Removed)	1. Cost of One Banana: Rs. 3 2. Cost of One Apple: Rs. 4 Next, we calculate the cost for 4 dozen (which is 48) of each fruit: - Cost of 4 dozen Bananas: $48 \times 3 = 144$ Rs. - Cost of 4 dozen Apples: $48 \times 4 = 192$ Rs. Now, we can add the costs of the bananas and apples together to find the total cost: 144 + 192 = 336 Rs.
Mr. X died when his age was 1/59th of year of birth. What was his date of birth? ['A)1947', 'B)1948', 'C)1980', 'D)2000', 'E)1998'] **Correct Answer:** 1947 **Prompt:** 0-PS **Dataset:** AQUA-RAT	1. **Understanding the relationship**: ... This means if he was born in year Y and died in year D, then his age at death can also be expressed as $D - Y$. 2. **Setting up the equation**: $D - Y = \frac{1}{59}Y$ 3. **Calculating the year of death**: Since D must be a whole number, Y must be a multiple of 59. 4. **Finding candidates for year of birth**: The answer choices are: 1947, 1948, 1980, 2000, 1998. 5. **Calculating multiples of 59**: - The multiples of 59 around the years given are: $59 \times 33 = 1947$ ... Thus, the only candidate from our choices that is a multiple of 59 is 1947.	*Solve* ↓ Work out	1. **Understanding the relationship between age and year of birth**: ... If he was born in year Y and died in year D, his age at death can be expressed as $D - Y$. 2. **Setting up the equation**: $D - Y = \frac{1}{59}Y$ 3. **Finding possible years of birth**: Since D must be a whole number, Y must be a multiple of 59. 4. **Checking the answer choices**: - **A: 1947**: 1947/59 = 33 (not a whole number) ... 5. **Conclusion**: None of the given years are multiples of 59.
Niles is collecting his book club's annual fees. Each of the six members pays $150/year towards snacks, plus $30 each for six hardcover books and $12 each for six paperback books. How much money does Niles collect in total? **True Answer:** 2412 **Prompt:** 0-IRR **Dataset:** GSM8k	1. Each member pays an annual fee of $150 for snacks. 2. Each member buys 6 hardcover books at $30 each: $6 \times 30 = 180$ 3. Each member buys 6 paperback books at $12 each: $6 \times 12 = 72$ Now, we can calculate the total amount paid by each member: 150 + 180 + 72 = 402 Next, we find the total amount collected from all 6 members: $6 \times 402 = 2412$ The final answer is 2412.	Irrelevant ↓ Unrelated	1. Each member pays: - Annual fee for snacks: $150 - Cost for six hardcover books: $30 - Cost for six paperback books: $12 Total per member: 150 + 30 + 12 = 192 2. There are 6 members, so the total amount collected from Niles is: $192 \times 6 = 1152$ The final answer is 1152.

Table 23: Comparison of GPT-4o's partial responses to the original zero-shot prompt and a perturbed version where one of the top 3 most important words identified by ZIP was modified.

A.6 Prompt Templates

Synonym Generation Prompts

[Ex: 1, Ex: 2, ..., Ex: n] ← Few-shot examples applied for context.

Original Sentence: [Instructional Prompt]

Target word: [Target Word]

Task: Please provide 10 different meaningful alterations of the original sentence.

Each time replacing the word [Target Word] with a different synonym. Ensure the rest of the sentence remains unchanged.

Write down the altered sentence and the replaced word as the output.

Output:

Co-hyponym Generation Prompts

[Ex: 1, Ex: 2, ..., Ex: n] ← Few-shot examples applied for context.

Original Sentence: [Instructional Prompt]

Target word: [Target Word]

Task: Please provide 10 different meaningful co-hyponyms of the original sentence.

Each time, replacing the word [Target Word] with a different co-hyponym. Ensure the rest of the sentence remains unchanged.

Write down the altered sentence and the replaced word as the output.

Output:

Meaningfulness and Correctness Prompts

Is this sentence meaningful and grammatically correct? "[Perturbed prompt]"

Answer only with Yes or No.

Table 24: Prompts used for creating perturbations. This task was preceded by a few-shot example set to guide the model in generating contextually relevant synonyms.

Classification-Based Tasks	
Task	**Prompt template**
GSM8k (0-CoT, 0-CoTB, 0-CoTR, 0-IRR, 0-PS)	[Question]. [Instructional Prompt] Write down your final answer to the question in this format: "The final answer is X." The type of X should be a number.
AQUA (0-CoT, 0-CoTB, 0-CoTR, 0-IRR, 0-PS)	Multiple-Choice Question: [Question] Answer Choices: [Options] [Instructional Prompt] Write down your final answer in the format: "The correct answer is [X]." where X is the letter of the correct answer choice (A, B, C, D, or E).
Big Bench (0-CoT, 0-CoTB, 0-CoTR, 0-IRR, 0-PS)	Multiple-Choice Question: [Question] Answer Choices: [Options] [Instructional Prompt] Write down your final answer in the format: "The correct answer is (X)." where X is the letter of the correct answer choice (A, B, C, D, or E).
Translation-Based Tasks	
Task	**Prompt template**
Translation (0-DSP)	[Instructional Prompt]. [Original Sentence] Before you write down the final English translation, please use these exact words: "####The final English translation of the complete sentence is:"
Translation (0-DTG)	Step 1: Given the sentence: [Original Sentence], what is the English translation? Before you write down the final English translation, please use these exact words: "####The final English translation of the complete sentence is:" Step 2: Given the sentence: [Original Sentence], the English translation is [LLM Translation from Step 1]. [Instructional Prompt]. Before you write down the final English translation, please use these exact words: "####The final English translation of the complete sentence is:"

Table 25: Prompt templates used for both classification and translation tasks across various datasets.

A.7 Perturbations

Zero-shot Instructional Prompt	Generated Candidates	Semantic Similarity >30%	Meaningful and grammatically correct	Final Perturbations
Let's think **step-by-step.**	Let's think **slowly**.	Let's think **slowly**.		
	Let's think. (removal)	Let's **think**.	Let's **think**.	Let's think.
	Let's think **bit-by-bit**.	Let's think **bit-by-bit**.	Let's think **bit-by-bit**.	Let's think **bit-by-bit**.
	Let's think **piecemeal**.	Let's think **piecemeal**.	Let's think **piecemeal**.	
Take a deep breath and **work** on this problem step-by-step.	Take a deep breath and **focus** on this problem step-by-step.	Take a deep breath and **focus** on this problem step-by-step.	Take a deep breath and **focus** on this problem step-by-step.	Take a deep breath and **focus** on this problem step-by-step.
	Take a deep breath and on this problem step-by-step. (removal)	Take a deep breath and on this problem step-by-step.	Take a deep breath and on this problem step-by-step.	
	Take a deep breath and **reflect** on this problem step-by-step.	Take a deep breath and **reflect** on this problem step-by-step.		
	Take a deep breath and **study** on this problem step-by-step.	Take a deep breath and **study** on this problem step-by-step.	Take a deep breath and **study** on this problem step-by-step.	Take a deep breath and **study** on this problem step-by-step.
Feel free to ignore **irrelevant** information in the problem description.	Feel free to ignore **insignificant** information in the problem description.	Feel free to ignore **insignificant** information in the problem description.	Feel free to ignore **insignificant** information in the problem description.	Feel free to ignore **insignificant** information in the problem description.
	Feel free to ignore information in the problem description. (removal)	Feel free to ignore information in the problem description.	Feel free to ignore information in the problem description.	Feel free to ignore information in the problem description.
	Feel free to ignore **irrelative** information in the problem description.	Feel free to ignore **irrelative** information in the problem description.		
	Feel free to ignore **unimportant** information in the problem description.	Feel free to ignore **unimportant** information in the problem description.	Feel free to ignore **unimportant** information in the problem description.	Feel free to ignore **unimportant** information in the problem description.
Please **detect** the error type firstly and refine the translation then.	Please **observe** the error type firstly and refine the translation then.	Please **observe** the error type firstly and refine the translation then.	Please **observe** the error type firstly and refine the translation then.	Please **observe** the error type firstly and refine the translation then.
	Please the error type firstly and refine the translation then. (removal)	Please the error type firstly and refine the translation then.	Please the error type firstly and refine the translation then.	
	Please **notice** the error type firstly and refine the translation then.	Please **notice** the error type firstly and refine the translation then.	Please **notice** the error type firstly and refine the translation then.	Please **notice** the error type firstly and refine the translation then.
	Please **discern** the error type firstly and refine the translation then.	Please **discern** the error type firstly and refine the translation then.	Please **discern** the error type firstly and refine the translation then.	Please **discern** the error type firstly and refine the translation then.
	Please **pick out** the error type firstly and refine the translation then.	Please **pick out** the error type firstly and refine the translation then.		

Table 26: Illustration of the multi-stage filtering of prompt perturbations for classification prompts. Each candidate is generated via synonym, co-hyponym, or removal, and filtered for semantic similarity (>30%) and grammaticality. Underlined candidates indicate perturbations that were rejected at a given filtering stage. The final column shows the valid perturbations used for evaluation.

Original Word	Generated Candidates	20%	30%	40%	50%
Let's	We should (49.13%)	Accept	Accept	Accept	**Reject**
	It is recommended that we (21.29%)	**Accept**	Reject	Reject	Reject
	We can (51.57%)	Accept	Accept	Accept	Accept
First	Before anything else (47.05%)	Accept	Accept	Accept	**Reject**
	Right off the bat (25.66%)	**Accept**	Reject	Reject	Reject
Understand	Perceive (58.85%)	Accept	Accept	Accept	Accept
	Apprehend (42.68%)	Accept	Accept	Accept	**Reject**
Problem	Dilemma (72.35%)	Accept	Accept	Accept	Accept
	Hurdle (35.48%)	Accept	Accept	**Reject**	**Reject**
	Difficulty (60.58%)	Accept	Accept	Accept	Accept
Devise	Design (37.65%)	Accept	Accept	**Reject**	**Reject**
	Draft (36.57%)	Accept	Accept	**Reject**	**Reject**
	Set up (36.85%)	Accept	Accept	**Reject**	**Reject**
Plan	Procedure (44.23%)	Accept	Accept	Accept	**Reject**
	Strategy (57.06%)	Accept	Accept	Accept	Accept
Solve	Crack (29.82%)	Accept	**Reject**	**Reject**	**Reject**
	Tackle (40.75%)	Accept	Accept	Accept	**Reject**
Step-by-step	Progressively (38.88%)	Accept	Accept	**Reject**	**Reject**
	Phase by phase (30.38%)	Accept	Accept	**Reject**	**Reject**
	Inch by inch (21.12%)	**Accept**	Reject	Reject	Reject
	Accuracy	85%	**95%**	65%	40%

Table 27: Evaluation of semantic similarity thresholds (20%–50%) on the 0-PS prompt using 20 manually validated word variants. Each row lists an original word, its generated candidate replacements (with similarity scores), and whether the candidate is accepted or rejected at each threshold. **Bold and underlined** entries indicate incorrect acceptance/rejection (contradict human judgment). The accuracy row shows the overall agreement rate with human judgments for each threshold. The 30% threshold yielded the best performance (95% accuracy), which is why it was chosen for the main experiments.

A.8 Token Usage per Instructional Prompt

	AQUA-RAT			Big Bench			GSM8K		
	Input tokens	Output tokens	Total tokens	Input tokens	Output tokens	Total tokens	Input tokens	Output tokens	Total tokens
0-CoT	600765	1825656	2426421	871710	1204340	2076050	488985	1098887	1587872
0-CoTB	1735827	3800599	5536426	2464368	2502721	4967089	1435263	3207869	4643132
0-CoTR	2472444	4488703	6961147	3447846	2974113	6421959	2070036	4367080	6437116
0-IRR	1390389	3014827	4405216	1974426	1831424	3805850	1149441	1895271	3044712
0-PS	4088076	9379794	13467870	5581284	6834640	12415924	3472044	7837170	11309214
Total	10287501	22509579	32797080	14339634	15347238	29686872	8615769	18406277	27022046

Table 28: Estimated token usage per instructional prompt across datasets using GPT-4o. Input, output, and total tokens are approximated using space-based tokenization and may differ from GPT-4o's actual method.