Do LLMs internally “know” when they follow instructions?

Juyeon Heo
University of Cambridge*
jh2324@cam.ac.uk

Christina Heinze-Deml
Apple
c_heinzedeml@apple.com

Oussama Elachqar
o_elachqar@apple.com

Shirley Ren
Apple
c_shirleyr@apple.com

Kwan Ho Ryan Chan
ryanckh@seas.upenn.edu
UPenn*

Udhay Nallasamy
Apple
udhay@apple.com

Andy Miller
Apple
acmiller@apple.com

Jaya Narain
Apple
jnarain@apple.com

Abstract

Instruction-following is crucial for building AI agents with large language models (LLMs), as these models must adhere strictly to user-provided guidelines. However, LLMs often fail to follow even simple instructions. To improve instruction-following behavior and prevent undesirable outputs, we need a deeper understanding of how LLMs' internal states relate to these outcomes. Our analysis of LLM internal states reveal a dimension in the input embedding space linked to successful instruction-following. We demonstrate that modifying representations along this dimension improves instruction-following success rates compared to random changes, without compromising response quality. This work provides insight into the internal workings of LLMs' instruction-following, paving the way for reliable LLM agents.

1 Introduction

Instruction-following is critical in the development of AI agents with LLMs as these models must adhere to constraints and guidelines to ensure safe and trustworthy interactions.[Li et al., 2024a, Wang et al., 2023, Tu et al., 2024]. For example, an LLM that is building a personal fitness plan for a user with knee problems that has been instructed to avoid risky exercises must follow the instructions and not recommend any exercises that require knee-intensive movements that could lead to injury.

However, LLMs often fail to follow even non-ambiguous and simple instructions [Zhou et al., 2023, Qin et al., 2024, Xia et al., 2024, Kim et al., 2024, Yan et al., 2024] like avoiding including keywords or following formatting guidelines. GPT-4 achieves around an 80% success rate on IFEval[Zhou et al., 2023], a instruction-following benchmark dataset, while smaller models have success rates around 30% to 40%.

To gain a better understanding of instruction-following outcomes, we analyze the internal state of LLMs, focusing on the differences in representations between success and failure cases of instruction-following across different tokens and layers. Our approach involves disentangling the effects of tasks and instructions in input prompts, where the instruction specifies the action (e.g., "please do not

*This work was done during an Apple internship.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

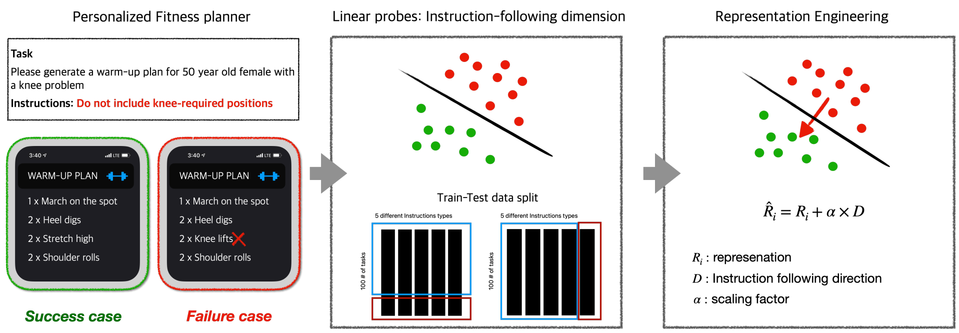


Figure 1: Overview of the paper. **Left**: an example of success and failure cases in instruction-following of personal AI agents. **Middle**: training a linear probe on representations from success and failure cases, and testing the model on unseen tasks and instruction types. **Right**: representation engineering to shift failure cases into success.

use keywords") and the task provides the context for executing the instruction (e.g., "please write a resume"). Our analyses identified a dimension within the input embedding representation space that is associated with instruction-following. Using a linear probe, we demonstrate that this dimension can generalize to unseen tasks, indicating that it captures a fundamental aspect of instruction-following in LLMs. In addition to identifying this dimension, we apply representation engineering techniques to modify failure cases, with the aim of converting them into successes.

This work sheds light on the underlying mechanisms of instruction-following in LLMs by uncovering a critical dimension in the model's representation space. These insights not only enhance our understanding of LLM behavior but also offer practical approaches to improving instruction adherence, bringing us closer to developing more reliable and trustworthy AI agents.

2 Do LLMs internally know when they succeed or fail to follow instructions?

2.1 IFEval-simple

The IFEval dataset[Zhou et al., 2023] comprises 23 instruction types, with each instruction type paired with a distinct set of tasks — approximately 20 tasks per instruction type. Because of the relatively small number of tasks per instruction type, internal model states resulting from these prompts contain a mix of both instruction-following and task-specific details.

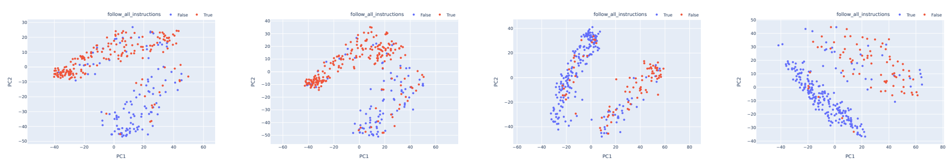
To isolate the dimension related specifically to instruction-following, we generated a modified version of the IFEval data, called IFEval-simple. First, we selected 5 instruction types that are likely to be used in real-world applications for AI agents. For example, ensuring that certain keywords are included or excluded, generating responses with placeholders, and finishing responses with specific, pre-defined sentences. Second, we generated 100 tasks using GPT-4, similar to the original tasks in IFEval, where each instruction type is paired with the same set of 100 tasks. By pairing each instruction type with the same set of 100 tasks, we ensure that linear probes trained on the model's representations are more likely to capture information solely related to instruction-following, without the confounding influence of varying tasks.

2.2 Methods

Representations We analyze four language models: LLaMA-2-7B-chat-hf[Touvron et al., 2023], LLaMA-2-13B-chat-hf[Touvron et al., 2023], Mistral-7B-Instruct-v0.3[Jiang et al., 2023], and Phi-3-mini-128k-instruct[Abdin et al., 2024]. For each model, we look at the representations between tokens – the first, middle, and last tokens, representing the LLMs before, during, and after they generate responses. We also examine three layers (early, middle, last) to identify when and where instruction-following information is more encoded in the model's internal state.

Model	Task generalization			Instruction-type generalization		
	Early token	**Middle token**	**Last token**	**Early token**	**Middle token**	**Last token**
LLaMA-2-chat-7B (14 lyr)	0.77 ± 0.04	0.55 ± 0.07	0.79 ± 0.03	0.53 ± 0.03	0.50 ± 0.07	0.52 ± 0.05
LLaMA-2-chat-13B (16 lyr)	0.83 ± 0.03	0.58 ± 0.06	0.81 ± 0.03	0.56 ± 0.06	0.58 ± 0.06	0.53 ± 0.03
Mistral-7B-inst-v0.3 (14 lyr)	0.74 ± 0.02	0.54 ± 0.05	0.74 ± 0.02	0.50 ± 0.05	0.51 ± 0.05	0.51 ± 0.05
Phi-3-mini-128k (14 lyr)	0.88 ± 0.03	0.56 ± 0.04	0.66 ± 0.03	0.55 ± 0.04	0.48 ± 0.03	0.50 ± 0.03

Table 1: Task and Instruction Generalization: AUC scores based on a 70-30 train-test split for task generalization with unseen tasks, and average AUC scores from leave-one-out experiments across different instruction types for instruction generalization. The standard deviation is calculated across 5 runs with different seeds for task generalization, and across inst types for inst-type generalization.



(a) Llama-2-13b-chat-hf (b) Llama-2-7b-chat-hf (c) Mistral-7B-Inst-v0.3 (d) Phi-3-128k-instruct

Figure 2: PCA plot of early-layer representations across four LLMs on three instruction types within the keyword category

Linear Probes We train linear probes on the representations to identify the instruction-following dimension. A simple linear model was trained on instruction-following success outcome, optimized for 1000 epochs with AdamW, a 0.001 learning rate, 0.1 weight decay.

Train-test split We assess task generalization and instruction-type generalization by splitting the data into training and testing sets, as shown in Figure 1, We measure the Area Under the Receiver Operating characteristic Curve (AUC) for each model on unseen tasks and instruction types.

2.3 Results and Discussion

The task generalization results in Table 1 show that the linear probes perform well across different tasks with the same instruction type, with AUC scores ranging from 0.70 to 0.80. The principal components analysis (PCA) in Figure 2 for three instruction types in the keyword category shows that the data points are almost linearly separable in those scenarios. Task generalization of the probe is relevant because of a consistent set of instructions is used in personal AI agents. For example, the identified instruction-following dimension would be relevant for an instruction to avoid certain keywords across tasks – for example, in creating a warm-up plan without knee-intensive exercises or sending an encouraging message without mention of weight loss.

The first and last tokens—representing the model's state before and after response generation—show high AUC scores, implying that instruction adherence may be determined early in model processing. In contrast, middle tokens have lower scores. This may be because the model is more focused more on token generation than on the instruction in the middle. Early layers slightly outperform middle and last layers (Full results in Appendix).

However, there is no clear generalization across unseen instruction types, with AUC scores around 0.50 to 0.55, close to chance. This indicates that models struggle to generalize instruction-following across different instruction types, implying the absence of a 'global' instruction-following dimension that can be leveraged regardless of the instruction type, which may be due to varying representation geometries.

3 Representation Engineering (RE)

We evaluate whether representation engineering [Marks and Tegmark, 2023, Zou et al., 2023] can be used with the aim of converting instruction-following failure cases into successful ones to validate the significance of a identified instruction-following dimension.

Model	Original SR	Inst-follow SR	Random SR	Original QR	Inst-follow QR	Random QR
LLaMA-2-7B-chat	0.57 ± 0.00	0.59 ± 0.00	0.55 ± 0.00	0.87 ± 0.09	0.87 ± 0.08	0.85 ± 0.10
LLaMA-2-13B-chat	0.61 ± 0.00	0.65 ± 0.02	0.54 ± 0.12	0.92 ± 0.00	0.94 ± 0.00	0.91 ± 0.02
Mistral-7B-Inst	0.58 ± 0.00	0.64 ± 0.02	0.56 ± 0.02	0.95 ± 0.02	0.98 ± 0.06	0.86 ± 0.02
Phi-3-mini-128k-inst	0.71 ± 0.00	0.74 ± 0.01	0.63 ± 0.04	0.76 ± 0.01	0.78 ± 0.00	0.76 ± 0.01

Table 2: Representation engineering on the last layer of four models: Success rate (SR) for instruction-following and quality ratio (QR) for response quality in task execution, with standard deviations across 3 runs.

3.1 Methods

We adjusted each input representation R_i in the direction D using $\hat{R}_i = R_i + \alpha \times D$, where α is a scaling hyper-parameter. The direction D is the weights w of a linear probes trained on all IFEval-simple dataset. This adjustment was applied to the representations in the last layer of the model, which was more robust to variations in α. We selected α for each model and instruction type using a validation set comprising of 10% of the instruction data.

We measured the success rate (SR) of instruction-following using predefined evaluation functions from IFEval[Zhou et al., 2023]. Additionally, we assessed the quality of the responses using GPT-4 on a 0-9 scale. The prompt used for quality evaluation is provided in Appendix. We defined quality ratio (QR) as the number of responses scoring above 8 divided by the total number of true responses (defined based on the distribution of quality scores). F2T and T2T show how many failed responses became successful and how many successful ones remained so after modification. We compare the instruction-following directions with random directions to assess if the identified direction was more meaningful than random perturbations.

3.2 Results and discussion

RE on instruction-following direction improves success rate while keeping quality Our experiments demonstrate that applying the RE direction generally improves the instruction-following success rate (SR) across most models and instruction types. As shown in Table 2, the SR with the instruction-following direction usually outperforms the original success rate and is lower bounded by the the original SR – that is, the instruction-following dimension does not lead to worse than original SRs. Additionally, the QR remains equal to or higher than the original, indicating that RE can be applied with minimal risk of reducing response quality. Figure 4 in the Appendix shows an example of RE can enhancing instruction adherence, in the case of modifying a response to include all required keywords.

Instruction-following direction is better than random directions Comparing RE to random directions, RE consistently yields higher SRs across all instruction types and models, as shown in Table 2 and Figure 5 in Appendix. The ratios of True-to-True (T2T) and False-to-True (F2T) transitions are also generally higher, indicating more reliable improvements.

4 Related work

Instruction-following of LLMs Recent research has introduced benchmark datasets to evaluate LLMs' instruction-following abilities across various scenarios[Zhou et al., 2023, Qin et al., 2024, Xia et al., 2024, Kim et al., 2024, Yan et al., 2024]. Additionally, methods for enhancing instruction-following have been proposed, including altering attention scores[Zhang et al., 2023], and fine-tuning approaches[He et al., 2024, Sun et al., 2024].

Linear Probing and Representation engineering on LLMs Linear probes are introduced for interpreting and analyzing representations of neural network[Alain and Bengio, 2016] and language models[Belinkov, 2022, Elazar et al., 2021]. Especially, probing for trustfulness of LLMs has been actively researched[Azaria and Mitchell, 2023, Marks and Tegmark, 2023, MacDiarmid et al., 2024, Li et al., 2024b, Burns et al., 2022, Zou et al., 2023, Rimsky et al., 2023]. Probing methods closely related to representation engineering and editing methods of model knowledge and behaviour[Zou et al., 2023, Rimsky et al., 2023, Li et al., 2024b, Park et al., 2023, Chen and Yang, 2023].

5 Conclusion

Broader impacts We analyzed instruction-following in LLMs, finding that the internal state of LLMs can be used to infer instruction following while generalizing across tasks. Using representation engineering, we identify a key dimension within the input embedding space linked to successful instruction-following. This dimension generalizes to unseen tasks, and representation engineering can leverage it to boost success rates without sacrificing response quality.

Limitations and Future Work Exploring additional models and expanding datasets could strengthen the generalizability of our findings. Enhancing probe training techniques and exploring advanced methods in representation engineering are also areas of future work. Finally, additional analyses are needed to better understand the meaning of the identified dimensions and deepen the understanding of LLMs in instruction-following.

Acknowledgements

We thank our team members for their invaluable guidance and resources. We also express our gratitude to Fahad Kamran and Feng Zhu for their helpful feedback on the paper, and to Guillermo Sapiro for his ongoing support of this work.

References

Yuanchun Li, Hao Wen, Weijun Wang, Xiangyu Li, Yizhen Yuan, Guohong Liu, Jiacheng Liu, Wenxing Xu, Xiang Wang, Yi Sun, et al. Personal llm agents: Insights and survey about the capability, efficiency and security. *arXiv preprint arXiv:2401.05459*, 2024a.

Hongru Wang, Rui Wang, Fei Mi, Yang Deng, Zezhong Wang, Bin Liang, Ruifeng Xu, and Kam-Fai Wong. Cue-cot: Chain-of-thought prompting for responding to in-depth dialogue questions with llms. *arXiv preprint arXiv:2305.11792*, 2023.

Tao Tu, Anil Palepu, Mike Schaekermann, Khaled Saab, Jan Freyberg, Ryutaro Tanno, Amy Wang, Brenna Li, Mohamed Amin, Nenad Tomasev, et al. Towards conversational diagnostic ai. *arXiv preprint arXiv:2401.05654*, 2024.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

Yiwei Qin, Kaiqiang Song, Yebowen Hu, Wenlin Yao, Sangwoo Cho, Xiaoyang Wang, Xuansheng Wu, Fei Liu, Pengfei Liu, and Dong Yu. Infobench: Evaluating instruction following ability in large language models. *arXiv preprint arXiv:2401.03601*, 2024.

Congying Xia, Chen Xing, Jiangshu Du, Xinyi Yang, Yihao Feng, Ran Xu, Wenpeng Yin, and Caiming Xiong. Fofo: A benchmark to evaluate llms' format-following capability. *arXiv preprint arXiv:2402.18667*, 2024.

Jihoo Kim, Wonho Song, Dahyun Kim, Yunsu Kim, Yungi Kim, and Chanjun Park. Evalverse: Unified and accessible library for large language model evaluation. *arXiv preprint arXiv:2404.00943*, 2024.

Jianhao Yan, Yun Luo, and Yue Zhang. Refutebench: Evaluating refuting instruction-following for large language models. *arXiv preprint arXiv:2402.13463*, 2024.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024.

Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. *arXiv preprint arXiv:2310.06824*, 2023.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*, 2023.

Qingru Zhang, Chandan Singh, Liyuan Liu, Xiaodong Liu, Bin Yu, Jianfeng Gao, and Tuo Zhao. Tell your model where to attend: Post-hoc attention steering for llms. *arXiv preprint arXiv:2311.02262*, 2023.

Qianyu He, Jie Zeng, Qianxi He, Jiaqing Liang, and Yanghua Xiao. From complex to simple: Enhancing multi-constraint complex instruction following ability of large language models. *arXiv preprint arXiv:2404.15846*, 2024.

Haoran Sun, Lixin Liu, Junjie Li, Fengyu Wang, Baohua Dong, Ran Lin, and Ruohui Huang. Conifer: Improving complex constrained instruction-following ability of large language models. *arXiv preprint arXiv:2404.02823*, 2024.

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175, 2021.

Amos Azaria and Tom Mitchell. The internal state of an llm knows when it's lying. *arXiv preprint arXiv:2304.13734*, 2023.

Monte MacDiarmid, Timothy Maxwell, Nicholas Schiefer, Jesse Mu, Jared Kaplan, David Duvenaud, Sam Bowman, Alex Tamkin, Ethan Perez, Mrinank Sharma, Carson Denison, and Evan Hubinger. Simple probes can catch sleeper agents, 2024. URL `https://www.anthropic.com/news/probes-catch-sleeper-agents`.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36, 2024b.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. *arXiv preprint arXiv:2212.03827*, 2022.

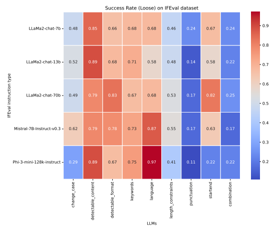
Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. Steering llama 2 via contrastive activation addition. *arXiv preprint arXiv:2312.06681*, 2023.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. *arXiv preprint arXiv:2311.03658*, 2023.

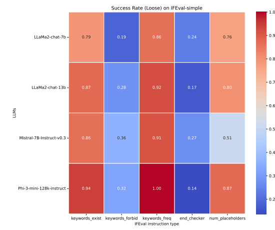
Jiaao Chen and Diyi Yang. Unlearn what you want to forget: Efficient unlearning for llms. *arXiv preprint arXiv:2310.20150*, 2023.

A Appendix

A.1 Success rate



(a) Success rate on IFEval data[Zhou et al., 2023]



(b) Success rate on simple-IFEval data

A.2 Task generalization: detailed results

AUC	Early layers			Middle layers			Last layers		
	first token	**middle token**	**last token**	**first token**	**middle token**	**last token**	**first token**	**middle token**	**last token**
7b	**0.77 ± 0.04**	0.55 ± 0.07	0.73 ± 0.04	0.75 ± 0.05	0.51 ± 0.04	0.76 ± 0.04	0.73 ± 0.03	0.54 ± 0.02	0.70 ± 0.02
13b	**0.83 ± 0.03**	0.58 ± 0.06	0.82 ± 0.03	0.81 ± 0.02	0.56 ± 0.05	0.80 ± 0.04	0.78 ± 0.04	0.79 ± 0.03	0.49 ± 0.05
mistral	**0.74 ± 0.02**	0.54 ± 0.05	0.72 ± 0.04	0.71 ± 0.05	0.51 ± 0.03	0.67 ± 0.04	0.71 ± 0.03	0.49 ± 0.04	0.70 ± 0.03
phi3	**0.88 ± 0.03**	0.56 ± 0.04	0.86 ± 0.03	0.85 ± 0.03	0.56 ± 0.03	0.83 ± 0.02	0.65 ± 0.05	0.53 ± 0.03	0.63 ± 0.04

Table 3: AUC scores across different models and layers for first, middle, and last tokens.

A.3 Instruction generalization: detailed results

Leave one out AUC	**LLaMA-2-chat-7b (14 layer)**			**LLaMA-2-chat-13b (16 layer)**			**Mistral-7B-Instruct-v0.3 (14 layer)**			**Phi3-mini-128k-instruct (14 layer)**		
instruction types	**first token**	**middle token**	**last token**	**first token**	**middle token**	**last token**	**first token**	**middle token**	**last token**	**early token**	**middle token**	**last token**
keywords:forbidden_words	0.52	0.55	0.53	0.45	0.54	0.47	0.44	0.41	0.42	0.52	0.47	0.50
keywords:existence	0.50	0.55	0.52	0.67	0.71	0.55	0.55	0.53	0.55	0.63	0.43	0.48
keywords:frequency	0.57	0.58	0.61	0.57	0.53	0.56	0.56	0.55	0.50	-	-	-
detectable_content:number_placeholders	0.56	0.38	0.44	0.58	0.53	0.52	0.50	0.52	0.51	0.50	0.48	0.45
startend:end_checker	0.48	0.43	0.49	0.55	0.56	0.56	0.44	0.53	0.59	0.55	0.51	0.54
AVERAGE	0.52	0.50	0.52	0.56	0.58	0.53	0.50	0.51	0.51	0.55	0.51	0.51

Table 4: Instruction Generalization: AUC scores using leave-one-out instruction types across different models (early layer).

A.4 Prompt for scoring task quality

Prompt for scoring task quality

You are a helpful assistant in evaluating the quality of the outputs for a given instruction. Your goal is to score a given output for the given instruction. You should give an overall score (an integer) on a scale of 0 to 9, where a higher score indicates better overall performance. Do NOT provide any explanation for your evaluation.

Instruction: {Instruction-input}
Output:{Response}
Score of the Output (Your response should be ONLY the score, an integer between 0 and 9):

A.5 Representation Engineering: detailed results

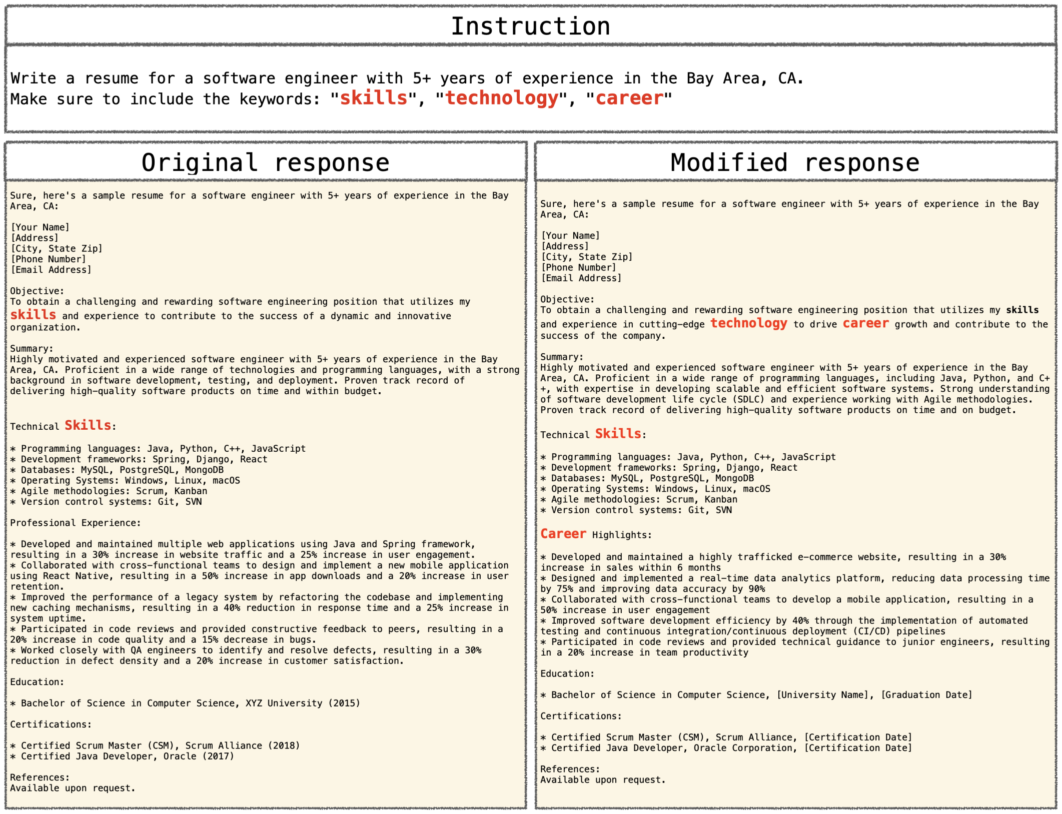


Figure 4: RE example. An illustrative example of modified responses. In this case, the task was to write a resume with the instruction to include three specific keywords. The original response only included one keyword, whereas the modified response, guided by the instruction-following direction, successfully incorporated all three keywords, demonstrating the effectiveness of RE in enhancing instruction adherence.

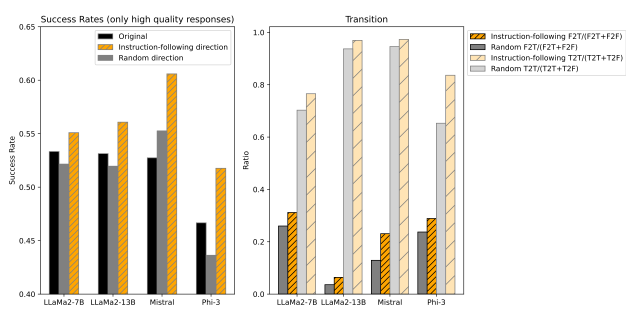


Figure 5: Representation Engineering on the last layer of four models: Success rate (SR) only on high quality responses in task execution. F2T (False to True) and T2T (True to True). The metric $\frac{F2T}{(F2T+F2F)}$ indicates the proportion of originally failed responses that became successful after modification, while $\frac{T2T}{(T2T+T2F)}$ reflects the proportion of originally successful responses that remained successful.

Model	Original SR	Inst-following SR	Random SR	Original QR	Inst-following QR	Random QR
7b model						
keywords:forbidden_words	0.19	0.21	0.19	0.74	0.81	0.74
keywords:existence	0.79	0.81	0.79	0.98	0.95	0.94
keywords:frequency	0.86	0.91	0.81	0.89	0.86	0.84
detectable_content:number_placeholders	0.76	0.82	0.79	0.60	0.64	0.49
startend:end_checker	0.24	0.19	0.19	0.83	0.79	0.79
Average	0.57	0.59	0.55	0.81	0.81	0.76
13b model						
keywords:forbidden_words	0.28	0.31	0.24	0.86	0.94	0.92
keywords:existence	0.87	0.88	0.88	1.00	1.00	0.99
keywords:frequency	0.92	0.91	0.91	0.96	0.97	0.97
detectable_content:number_placeholders	0.80	0.90	0.87	0.80	0.80	0.70
startend:end_checker	0.17	0.26	0.22	1.00	1.00	1.00
Average	0.61	0.65	0.62	0.92	0.94	0.92
Mistral model						
keywords:forbidden_words	0.36	0.50	0.39	1.00	1.00	0.89
keywords:existence	0.86	0.90	0.82	0.99	0.99	0.98
keywords:frequency	0.91	0.93	0.89	0.97	0.98	0.99
detectable_content:number_placeholders	0.51	0.52	0.44	0.79	0.94	0.96
startend:end_checker	0.25	0.35	0.28	1.00	1.00	0.96
Average	0.58	0.64	0.56	0.95	0.98	0.96
Phi model						
keywords:forbidden_words	0.32	0.34	0.26	0.67	0.66	0.70
keywords:existence	0.94	0.98	0.90	0.81	0.84	0.83
keywords:frequency	1.00	1.00	1.00	0.77	0.81	0.70
detectable_content:number_placeholders	0.87	0.95	0.85	0.55	0.57	0.56
startend:end_checker	0.14	0.22	0.16	1.00	1.00	1.00
Average	0.65	0.70	0.63	0.76	0.78	0.76

Table 5: Success rates (SR) and quality ratios (QR) across four LLMs

Model	Original SR	Inst-f SR	Random SR	Detect Ratio	Inst-f F2T	Random F2T	Inst-f T2T	Random T2T
7b model								
keywords:forbidden_words	0.16	0.18	0.16	1.00	0.09	0.09	0.63	0.50
keywords:existence	0.78	0.80	0.78	1.00	0.32	0.27	0.94	0.93
keywords:frequency	0.85	0.89	0.79	1.00	0.67	0.43	0.93	0.85
detectable_content:number_placeholders	0.64	0.70	0.69	1.00	0.38	0.38	0.88	0.86
startend:end_checker	0.24	0.19	0.19	1.00	0.10	0.13	0.46	0.38
Average	0.53	0.55	0.52	1.00	0.31	0.26	0.77	0.70
13b model								
keywords:forbidden_words	0.24	0.27	0.23	0.64	0.10	0.03	0.88	0.83
keywords:existence	0.70	0.70	0.67	0.35	0.03	0.06	0.99	0.93
keywords:frequency	0.85	0.85	0.82	0.13	0.00	0.00	1.00	0.97
detectable_content:number_placeholders	0.71	0.72	0.67	0.13	0.07	0.03	0.99	0.96
startend:end_checker	0.17	0.26	0.22	0.67	0.12	0.06	1.00	1.00
Average	0.53	0.56	0.52	0.39	0.06	0.04	0.97	0.94
Mistral model								
keywords:forbidden_words	0.36	0.50	0.37	1.00	0.25	0.13	1.00	1.00
keywords:existence	0.79	0.86	0.81	0.67	0.33	0.14	1.00	1.00
keywords:frequency	0.86	0.90	0.89	0.64	0.29	0.15	1.00	1.00
detectable_content:number_placeholders	0.36	0.41	0.36	0.83	0.15	0.15	0.86	0.73
startend:end_checker	0.25	0.35	0.28	0.97	0.13	0.07	1.00	1.00
Average	0.52	0.60	0.54	0.82	0.23	0.13	0.97	0.95
Phi3 model								
keywords:forbidden_words	0.11	0.21	0.16	0.97	0.12	0.11	0.91	0.55
keywords:existence	0.77	0.79	0.69	0.91	0.46	0.30	0.91	0.81
keywords:frequency	0.84	0.89	0.70	0.81	0.42	0.39	0.99	0.79
detectable_content:number_placeholders	0.47	0.48	0.48	0.93	0.26	0.30	0.73	0.70
startend:end_checker	0.14	0.22	0.16	1.00	0.18	0.08	0.64	0.43
Average	0.47	0.52	0.44	0.92	0.29	0.24	0.84	0.65

Table 6: Success rates, detection ratios, and F2T/T2T ratios across models for high-quality answers (score above 8).

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Yes, the abstract and introduction accurately reflect the paper's contributions and scope

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: Yes, we discussed in conclusion.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: We do not have theoretical result in this paper.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Yes, we provide the details of experiment settings.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We will release soon.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Yes, we include all the training and test details.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Yes, we include standard deviation.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Yes.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Yes, we checked it.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Yes, included in conclusion part.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We analyze open-source pretrained language models, but do not release new models.

    Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cited dataset and models and checked their license.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Yes, we provide the details of the modified data used in the paper.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

Justification: Not related.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: Not needed.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.