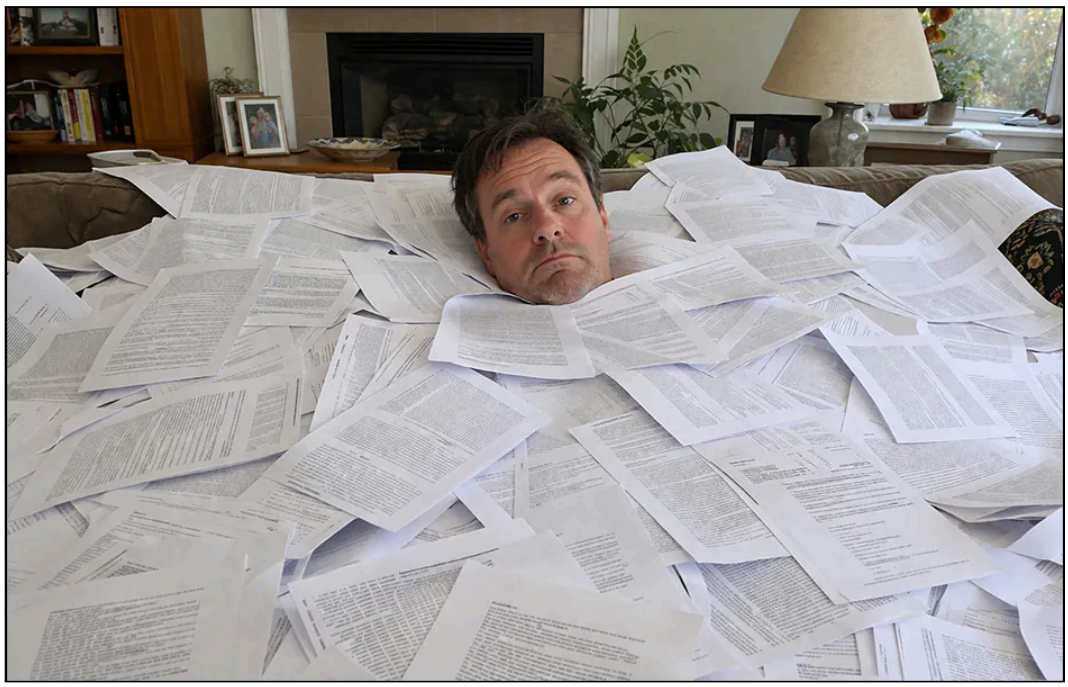
Verbosity Decreases Accuracy in Large Language Models

Originally published at https://www.unite.ai/verbosity-decreases-accuracy-in-large-language-models/



New research finds that forcing Large Language Models to give shorter answers notably improves the accuracy and quality of their answers.

Anyone who has tried to stop a chatbot from 'rambling' will recognize the conclusions of new research: forcing AI to give shorter replies makes it more accurate.

Investigating the reasons why larger AI chatbots perform worse in this respect than smaller ones, in certain cases (known as inverse scaling), the research found that forcing 31 popular Large Language Models (LLMs) to give *shorter* answers caused up to a 26.3% improvement in the accuracy of their answers:

> *'Results provide compelling causal evidence: brevity constraints improved large model accuracy by 26.3 percentage points and reduced the inverse scaling gap by 67% (from 44.2% to 14.8%, paired t-test: t = 7.80, p < 0.0001).'*

Excessive verbosity is a frequent complaint among end-users, not least among those who use commercial-grade models such as ChatGPT, where the support forums feature this topic frequently.

The domain most affected by remedying verbosity in responses is *mathematics*, where the AIs tested were constrained to answer in 50 words or less. For *reading comprehension* tasks, they were restricted to a mere 10 words in their answers.

The paper defines AI's tendency towards verbosity as *overthinking*, wherein the central message is not just obscured by verbiage, but sometimes even adversely affected by it. The smaller the model is, the paper observes, the less this remedy is needed, or works.

The research concludes that there is nothing *architectural* that needs addressing in order to apply this solution systematically. However, in a user's chat session, a directive towards brevity would likely need repeating, whereas a globally-applied system prompt – which would need implementing as an engineering default on platforms such as ChatGPT – could make shorter answers default behavior.

Blustery Winds

None of this explains exactly why larger models tend towards verbosity, since this is something that affects open source models as well. The paper suggests that protocols and common practices in Reinforcement Learning from Human Feedback (RLHF) techniques could offer an explanation*:

> *'A plausible origin is RLHF alignment training, where human annotators reward thoroughness disproportionately in larger models with greater capacity to act on length-reward signals–consistent with verbosity differences being larger in instruction-tuned than base model variants.*
>
> *'Prior work documents systematic length bias in reward models, where annotators conflate length with quality.*
>
> *'Larger models, having greater capacity to satisfy length-reward signals, may internalize verbose generation more deeply than smaller models, producing the scale-dependent overthinking we observe.'*

In humans, verbosity may occur to fill in a silence, or to mask feelings of awkwardness, because of mental illness, or to conceal a lack of knowledge. In effect, an AI could only be influenced by these factors through absorption of training data that reflects/manifests these traits.

In dataset corpora, other motivations exist for prolix answers, such as the SEO incentive to produce longer text content, for instance in recipe posts, wherein length becomes (often erroneously) associated with authority.

What can't be entirely discounted is that API-based platforms, incentivized to nudge users to a higher and more expensive subscription tier, either encourage or do not superintend verbosity, since it increases token use quite cheaply, without the need for excessive reasoning or RAG calls[†].

The new paper is titled *Brevity Constraints Reverse Performance Hierarchies in Language Models*, and comes from the Department of Computer Science at the Sweden Polytechnic Institute in Chattogram, in Bangladesh.

Method

To test the paper's theories, 31 language models were evaluated – too many to list in text-form here, but depicted in the image below:

LLaMA-2-13B	LLaMA-3.1-Minitron-4B-Depth-Base	Gemma-2-2B-IT	Phi-3-Mini-4K-Instruct
LLaMA-3-70B-Base	LLaMA-3.1-Minitron-4B-Width-Base	Gemma-2-9B-IT	Phi-3.5-Mini-Instruct
LLaMA-3-70B-Instruct	LLaMA-3.1-Nemotron-Nano-8B-V1	Gemma-3-1B-IT	StableLM-2-1.6B
LLaMA-3.1-8B-Instruct	Qwen2.5-0.5B-Instruct	Gemini-2.0-Flash	StableLM-Zephyr-3B
LLaMA-3.1-405B-Instruct	Qwen2.5-3B-Instruct	Mistral-7B-Instruct-v0.3	Yi-1.5-6B-Chat
LLaMA-3.2-1B-Instruct	Qwen2.5-7B-Instruct	Mistral-Small-24B-2501	Kimi-K2-32B-Instruct
LLaMA-3.2-3B-Instruct	Qwen2.5-14B-Instruct	DeepSeek-67B-Base	GPT-OSS-20B
LLaMA-3.3-70B-Versatile	Qwen2.5-32B-Instruct	DeepSeek-LLM-7B-Base	

The Large Language Models (LLMs) tested in various parts of the trials for the new paper.

The models were evaluated against five benchmark collections: GSM8K, for mathematical reasoning; BoolQ, for reading comprehension; CommonsenseQA, for commonsense reasoning; ARC-Easy, covering science questions; and MMLU-STEM, also for scientific knowledge.

Answers were generated using greedy decoding to ensure deterministic outputs, then extracted with task-specific rules, with accuracy measured as the share of correct responses against ground truth.

Models were split by size, with those at or below ten billion parameters treated as 'small', and those above seventy billion as 'large', based on observed performance gaps.

Inverse scaling was quantified by comparing the performance of these groups on each problem, marking cases where smaller models outperformed larger ones; statistical effect size was then used to confirm that these gaps reflected consistent, meaningful differences rather than noise.

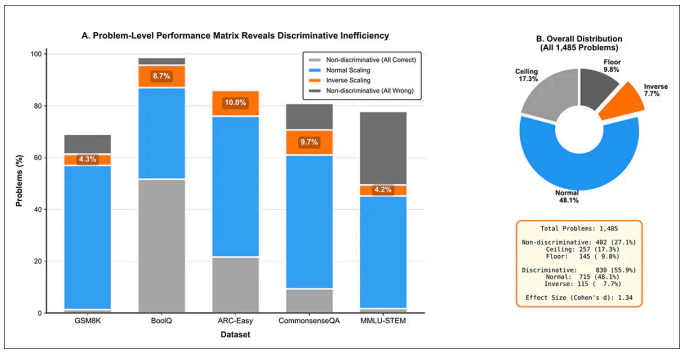
Ruling Out Recall

To discount the possibility that models were simply recalling training data, three separate checks were carried out, examining how varied the answers were; how much their length fluctuated; and how errors were made. If models were relying on memorized patterns, responses would tend to repeat or follow fixed templates; instead, answers proved mostly unique across models, with noticeable variation in length, from one response to another.

Errors were also inspected directly, with most failures taking the form of *long, incorrect explanations*, rather than short, evasive answers – indicating that the models were generating actual reasoning. rather than retrieving stored responses.

Data and Tests

Testing for individual questions rather than headline scores, a large share of benchmark tasks turned out to be uninformative, with 27.1% failing to separate models at all because either every system succeeded or every system failed, leaving no real signal about relative performance:



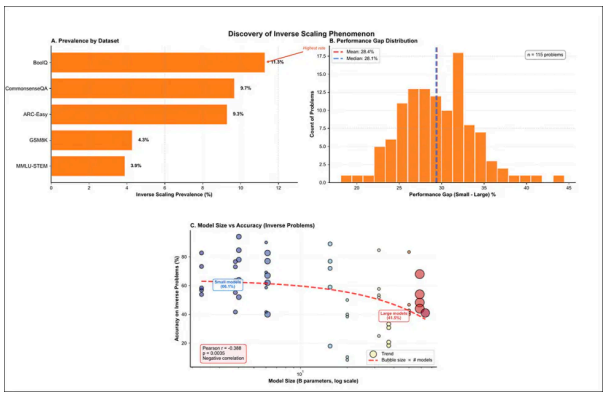
Problem-level breakdown across five benchmarks showed that a substantial share of tasks failed to distinguish between models, while a smaller but consistent portion exhibited inverse scaling, where smaller models outperformed larger ones. The overall distribution across 1,485 problems indicates that 7.7% exhibit inverse scaling. Source

Among the questions that *did* differentiate models, most behaved as expected, with larger systems performing better, but a smaller group showing the opposite pattern, with smaller models coming out ahead. Across all problems, inverse scaling appeared in 7.7% of cases, indicating that the effect is not marginal.

Inverse Scaling Surfacing

Across the five benchmarks, 115 problems were found where smaller models outperformed larger ones, making up 7.7% of all 1,485 tasks, indicating that inverse scaling is not some kind of rare or marginal occurrence in this context.

In fact, the effect showed up in *every* dataset: most strongly in BoolQ, and more weakly in CommonsenseQA, ARC-Easy, GSM8K, and MMLU-STEM, indicating that it is widespread, but varies by task:



Inverse scaling appeared across all benchmarks, ranging from 3.9% in MMLU-STEM to 11.3% in BoolQ, with 115 problems in total. Performance gaps favored smaller models by an average of 28.4 percentage points. Accuracy declined as model size increased, with small models reaching 66.1%, compared to 41.5% for larger ones.

The size of the gap proved substantial, with smaller models ahead by an average of 28.4 percentage points, and every case showing the same direction of advantage, pointing to a consistent drop in performance, rather than occasional or *ad hoc* errors.

The same pattern held across different model families, including Llama, Qwen, Gemma, and Mistral, where larger versions underperformed smaller ones, with accuracy tending to fall as model size increased on these problems.

The gap between small and large models was large enough that chance would be unlikely to explain it; and because the same pattern appeared across different benchmarks, tasks, and model families, inverse scaling emerged as a consistent rather than random effect.

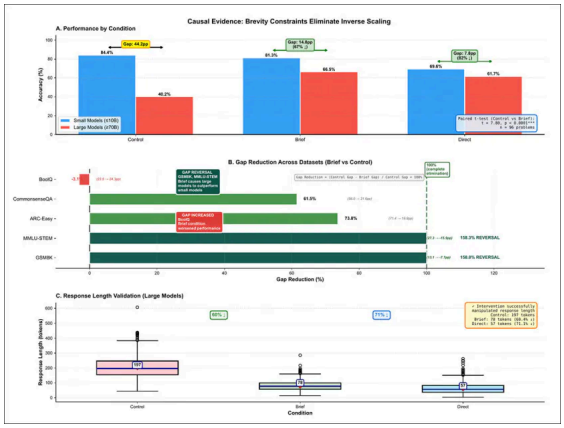
Looking within each model family, larger versions repeatedly performed worse than smaller ones on these problems, suggesting that the decline may be tied to scale itself, rather than to differences in design.

The results also point to a limit for each task, where increasing model size starts to hurt performance, showing that bigger models do not always lead to better results.

Examining Overthinking

Having established that larger models sometimes perform worse on certain domains, the analysis turned to why this happens, proposing that the problem is not lack of ability, but too much explanation – i.e., cases where longer answers begin to *obscure correct reasoning*.

Across the data, longer responses were linked to lower accuracy on these difficult problems, even though large and small models produced roughly similar numbers of reasoning steps, suggesting that the issue is not how much reasoning is done, but how it is expressed:



Shorter answers improved large model performance and reduced the gap with smaller models, cutting the difference from 44.2 percentage points to 14.8 (and in some cases reversing it entirely), while direct answer formats narrowed it further. The strongest gains appeared in GSM8K and MMLU-STEM, where rankings flipped in favor of larger models, and response-length checks confirmed that the intervention worked, with outputs dropping from around 197 tokens to under 80, linking reduced verbosity to improved accuracy.

When answers were forced to be shorter, large models improved sharply, reducing a considerable performance gap, and, in some cases, nearly eliminating it. Conversely, small models changed very little, indicating that verbosity was *actively harming* larger systems.

The effect proved to vary by task, with some benchmarks benefiting strongly from shorter answers, and others requiring a degree of explanation; but in several cases, the ranking between small and large models completely reversed once verbosity was constrained, revealing that larger models had hidden capability *that was being masked by over-elaboration*.

Further analysis showed that large models tended to produce longer outputs overall, despite using slightly fewer explicit reasoning steps, indicating a more diffuse and less structured style of reasoning.

Conversely, smaller models gave shorter, more direct answers, suggesting that *the way reasoning is expressed*, rather than the amount of reasoning itself, drives the drop in performance.

The authors conclude in general that brevity constraints bring accuracy benefits, and consider that this could become a foundational feature in language models, rather than a repetitive, user-applied constraint that does not survive across sessions; and they state:

> *'[Brevity] constraints help large models dramatically while barely affecting small models.*
>
> *'If verbosity were incidental rather than causal, uniform accuracy changes would be expected across both size categories. The differential response confirms that overthinking is a scale-specific failure mode, not a task difficulty effect.'*

Conclusion

Beyond the open source versions tested by the researchers, the verbosity problem appears, anecdotally, with some frequency across many major models besides ChatGPT, including Claude, Gemini, and Grok.

Whether or not these platforms are overlooking the verbosity issue because it increases token usage and encourages higher spending†, it doesn't seem reasonable that they would accept the drop in accuracy that accompanies this 'nudging'.

It would be interesting to see if any empirical method could determine with certainty where the tendency towards verbosity comes from. Anyone who has ever tried to get a chatbot to actually chat, rather than 'blogging' verbose, multi-step answers at the user, will have realized that the model has been severely imprinted with 'all-in-one' user guides, and 'accepted solutions' in its training data.

It would therefore be very interesting to see if a model specifically trained on step-by-step conversations could actually steer away from the verbose, summarizing tendency. However, it seems likely that some constraints or filters would have to be placed on the weight that the model is trained to give on the 'decided' answer, so that it trains directly also on the preceding material – essentially, the explicit reasoning explicit in turn-based conversations.

Since apposite data of this kind seems likely to be rare, perhaps the only way forward with this approach would be via synthetic data, wherein 'compounded' final conclusions are picked apart conversationally – ironically, in much the same manner as the AI podcasts that Google NotebookLM can interpret from plain text input.

* *My conversion of the authors' inline citations to hyperlinks.*

† *But let's be honest, the shortfall between actual AI provision costs and subscription charges is currently so large that this would merely damage user-base reputation without solving the severe underlying economics of this phase of the 'conversion' and 'convergence' phase of AI.*

First published Sunday, April 5, 2026

Writer on machine learning, domain specialist in human image synthesis. Former head of research content at Metaphysic.ai, until its dissolution into DNEG's Brahma.ai.
Portfolio site: martinanderson.ai
Contact: martin@martinanderson.ai