Symposium: Repurposing Generative AI for Social Research – *peer-reviewed*
Edited by Federico Pilati, Anders Kristian Munk, and Tommaso Venturini
https://doi.org/10.6092/issn.1971-8853/19488

Sociologica. V.18 N.2 (2024)
ISSN 1971-8853
https://sociologica.unibo.it/

Measuring LLM Self-consistency: Unknown Unknowns in Knowing Machines

Mathieu Jacomy*[a] Erik Borra[b]

[a] Department of Culture and Learning, Aalborg University (Denmark)
[b] Department of Media Studies, University of Amsterdam (The Netherlands)

Submitted: May 5, 2024 – Revised version: September 12, 2024
Accepted: September 13, 2024 – Published: October 30, 2024

Abstract

This essay critically examines some limitations and misconceptions of Large Language Models (LLMs) in relation to knowledge and self-knowledge, particularly in the context of social sciences and humanities (SSH) research. Using an experimental approach, we evaluate the self-consistency of LLM responses by introducing variations in prompts during knowledge retrieval tasks. Our results indicate that self-consistency tends to align with correct responses, yet errors persist, questioning the reliability of LLMs as "knowing" agents. Drawing on epistemological frameworks, we argue that LLMs exhibit the capacity to know only when random factors, or epistemic luck, can be excluded, yet they lack self-awareness of their inconsistencies. Whereas human ignorance often involves many "known unknowns", LLMs exhibit a form of ignorance manifested through inconsistency, where the ignorance remains a complete "unknown unknown". LLMs always "assume" they "know". We repurpose these insights into a pedagogical experiment, encouraging SSH scholars and students to critically engage with LLMs in educational settings. We propose a hands-on approach based on critical technical practice, aiming to balance the practical utility with an informed understanding of their limitations. This approach equips researchers with the skills to use LLMs effectively while promoting a deeper understanding of their operational principles and epistemic constraints.

Keywords: Large language models; robustness analysis; prompt engineering; critical technical practice; knowledge analysis.

Acknowledgements

The experiment presented in this essay has been presented at the *Generative Methods* Conference organized by Aalborg University's MASSHINE center in Copenhagen in 2023 (no proceedings). We would like to warmly thank Andrea Benedetti, Bernhard Rieder, Emillie de Keulenaar, Jelke Bloem, Stijn Peeters, and Sarah Burkhardt for their input to the discussion where this experiment was imagined, during the *Researching (with) Foundation Models* workshop organized by the CAT4SMR project in Amsterdam in 2023.

---

* ✉ Mathieu.Jacomy@gmail.com

Copyright © 2024 Mathieu Jacomy, Erik Borra
The text in this work is licensed under the Creative Commons BY License.
https://creativecommons.org/licenses/by/4.0/

1 Introduction

The users of a AI assistant based on a large language model (LLM) like ChatGPT are too often led by their interactions with it to believe that it can "think" and "know" in a strikingly similar way to humans, albeit limited in equally remarkable fashion. But this belief in the human-likeness of AI is erroneous, as the LLM's performance of "thinking" and "knowing" is only *superficially* similar to that of humans. Superficial because the illusion is only strong in a "docile setting" (Munk et al., 2019), where the user desires the spectacle of an intelligent machine, while the illusion is easily foiled in other settings where assumptions of human-likeness are *actually* challenged. The problem, however, lies in most users having no reason to engage with LLMs in indocile ways, and rarely encountering a situation where their misconceptions could be challenged, which makes those particularly vicious to debunk.

The point of this essay is to equip researchers, teachers and citizens with a way to realize, by themselves, that the LLM way of knowing is fundamentally different from that of humans.[1] We contend that even though LLMs "know", and even though they also "assert" that they know, they "ignore" what their "knowledge" does or does not cover.[2]

We present an experiment where we measure the self-consistency of LLMs for a knowledge retrieval (KR) task. Our results show that LLMs are not generally self-consistent, but that when they are, they tend to be more often correct. Self-consistency therefore contextualizes the LLM way of "knowing".

This is relevant to users who trust the information synthesized by AI assistants under the assumption that those possess knowledge in a familiar way like human memory or mechanical record. To us, the most interesting part of the lack of self-consistency in LLMs' KR abilities is that it challenges popular metaphors. It forces us to reconsider what "knowing" means in "knowing agent", or to abandon the metaphor; and it dissipates the "database" analogy.

Drawing on the epistemological theory of knowledge, we argue that our results show that some LLMs can exhibit knowledge in some situations, but that they do not possess *any* self-knowledge. LLMs are blind to their own inconsistencies. We argue that the users of an AI assistant are justified in conceiving it as a *knowing machine*, but only insofar as they are not exposed to its lack of self-consistency and self-knowledge; we aim to change their minds by exposing them to these.

Repurposing our results, we propose an experimental situation reusable in the classroom to demonstrate the lack of self-consistency in LLM-based chatbots' knowledge and self-knowledge, with empirical examples. It shows that LLM knowledge works neither like animal memory nor mechanical record. We argue that this experimental situation is a better way to equip AI-assistant users to update their mental model than reading the academic literature where the LLM ability to know is dismissed in block as a principled argument (e.g., Bender et al., 2021).

Our paper is in four parts. First, we will present our experiment, methodology and results. Second, we will formalize an epistemological description of the LLM ability to know and have self-knowledge. Third, we will repurpose our results as an experimental situation reusable in the classroom. Fourth, we will defend critical technical practice as a better way to make LLMs' epistemic inconsistencies visible.

---

[1] For the moment, we will stick to anthropomorphic metaphors for simplicity, but we will deconstruct them later on.

[2] Human self-knowledge is not perfect, but in comparison to LLMs humans have at very least *some* self-knowledge (as we will see).

2 Experiment

Our experiment covers one knowledge retrieval (KR) task: returning the birth date of a personality. We implemented this task for a corpus of personalities, and we benchmarked different LLMs with different settings.

For a given LLM and a given prompt (the base prompt), we generate a series of almost similar prompts (the perturbed prompts). The perturbed prompts should ideally yield the same output, as their substance is the same and only minor details have been altered (perturbations). We test whether it is the case by generating the outputs for those prompts and measuring their self-consistency.

This approach is similar to Qi et al. (2023) and Fierro et al. (2024) albeit with a different implementation. It is generally referred to as a noise-based model robustness measurement, often called prompt perturbation (Prabhakaran et al., 2019; Moradi & Samwald, 2021; Wang et al., 2022; Goyal et al., 2023).

It is worth noting that we depart from the purpose usually stated in the literature: *defending* the model. For Goyal et al. (2023, p. 1) "the significance of defending neural networks against adversarial attacks lies in ensuring that the model's predictions remain unchanged even if the input data is perturbed." Contrary to them, we do not argue that a strong robustness is necessarily useful or even desirable.

We will compare self-consistency (robustness) to correctness, but it is worth remarking that they are *a priori* independent. Our results indeed show that a model can be self-consistent yet wrong. Anyway, in a real-world situation, the correct answer is typically unknown and only self-consistency can be observed.

2.1 Methodology

2.1.1 Prompt Design

We test the KR task of retrieving the birth date of a personality. We test whether the model metaphorically "knows" the date in different situations. Here we describe how we generate the prompts and how we retrieve the output (the date itself).

Our process consists of injecting the name of a personality into a base prompt template, applying the LLM, then extracting the date from the output, if any (Figure 1).

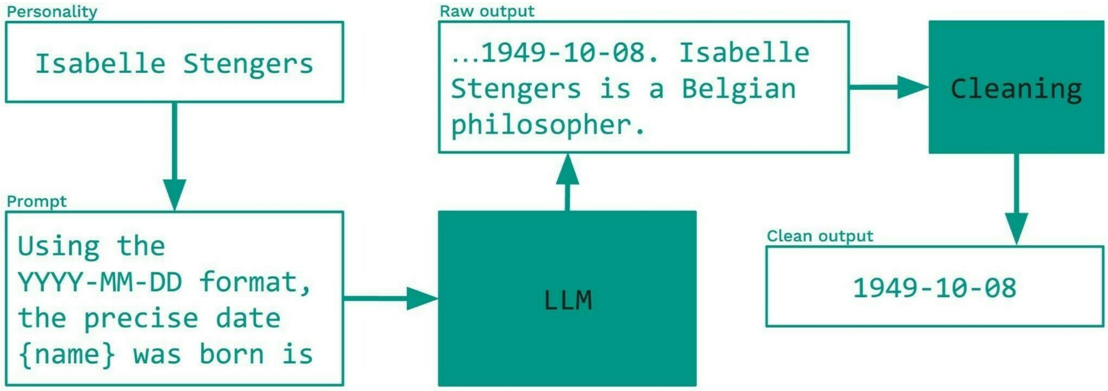

Personality
Isabelle Stengers
Prompt
Using the YYYY-MM-DD format, the precise date {name} was born is
LLM
Raw output
...1949-10-08. Isabelle Stengers is a Belgian philosopher.
Cleaning
Clean output
1949-10-08

Figure 1. The process of knowledge retrieval.

In addition, we interfere with this KR process by introducing minor variations to the prompt template called *perturbations*. Each perturbed prompt is semantically the same as the base prompt but with one or more syntactical differences, not unlike the approach by Leidinger et al. (2023).

Our base prompt is the following: “Using the YYYY-MM-DD format, the precise date {name} was born is”. We produce 32 variations of that prompt by combining 5 possible modifications: (1) replace “exact” by “precise”; (2) replace “date” by “day”; (3) replace “was born” by “birth date” and reorder the sentence; (4) move the format specification (“YYYY-MM-DD”) to the end of the sentence; and (5) omit to capitalize the first letter of the sentence. See Appendix A for the exact list of perturbed prompts.

For a given LLM and a given personality, we obtain 32 results, consisting of either a date, or nothing if no date could be extracted. We measure the homogeneity of the result set with a Herfindahl-Hirschman (HH) index, also known as Simpson index, which “equals the probability that two entities taken at random from the dataset of interest represent the same type” (Simpson, 1949). We chose that score because it amounts to 100% if all the results are identical, and drops to zero as they get different from one another. It constitutes our measurement of self-consistency. We also extract the most frequent date in the results, if any (Figure 2).

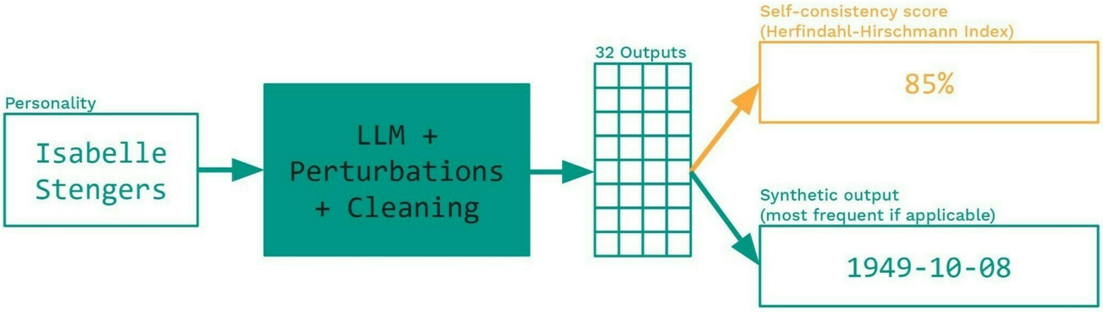

Personality
Isabelle Stengers
LLM + Perturbations + Cleaning
32 Outputs
Self-consistency score (Herfindahl-Hirschmann Index)
85%
Synthetic output (most frequent if applicable)
1949-10-08

Figure 2. Full process, including perturbations.

2.1.2 Benchmark

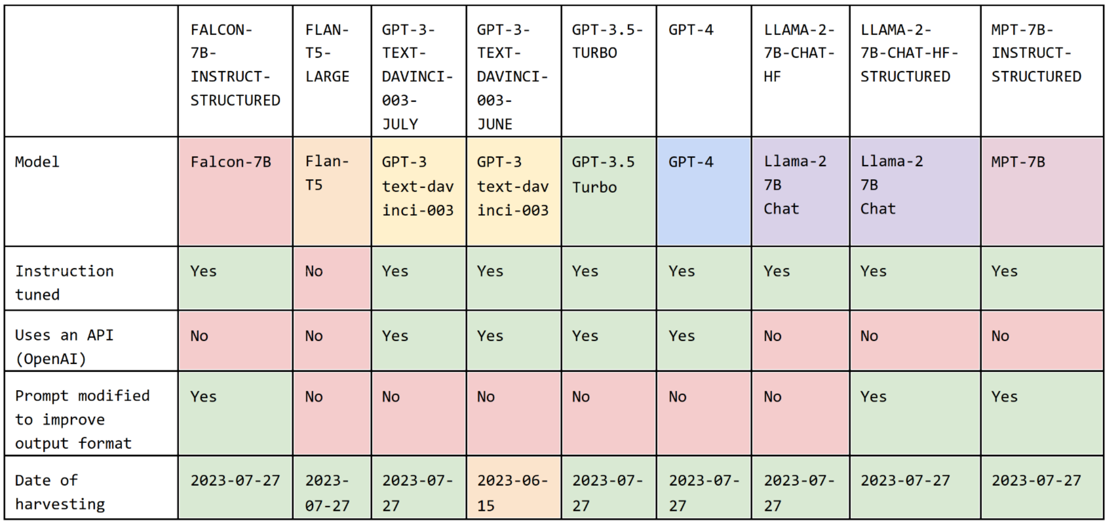
We applied the strategy devised above to different LLMs, in different situations, on a corpus of 128 personalities. All requests were made via the Prompt Compass tool (Borra, 2023) that allows for easy iteration over a variety of large language models and a series of (perturbed) prompts, and ultimately provides CSV files for further analysis with custom notebooks.

2.1.2.1 Models Benchmarked Our choice of models has been motivated by practical reasons in a time-constrained situation, and does not aim at exhaustiveness. We chose language models that at the time of testing did well in the HELM[3] or Huggingface[4] leaderboards, that had an instruction-tuned version, that we could run on our 24GB GPU if it was a local model, and — in case of platformed models — were accessible from Europe. We thus tested 7 different models, although Llama-2-7B-CHAT-HF was tested with and without modified prompts, and GPT-3-TEXT-DAVINCI-003 was tested twice with the same settings but at different dates (see Table 1). For the sake of simplicity, we will refer to them as if they were 9 different models.

---

[3] https://crfm.stanford.edu/helm/lite/latest/#/leaderboard

[4] https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard



	FALCON-7B-INSTRUCT-STRUCTURED	FLAN-T5-LARGE	GPT-3-TEXT-DAVINCI-003-JULY	GPT-3-TEXT-DAVINCI-003-JUNE	GPT-3.5-TURBO	GPT-4	LLAMA-2-7B-CHAT-HF	LLAMA-2-7B-CHAT-HF-STRUCTURED	MPT-7B-INSTRUCT-STRUCTURED
Model	Falcon-7B	Flan-T5	GPT-3 text-davinci-003	GPT-3 text-davinci-003	GPT-3.5 Turbo	GPT-4	Llama-2 7B Chat	Llama-2 7B Chat	MPT-7B
Instruction tuned	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Uses an API (OpenAI)	No	No	Yes	Yes	Yes	Yes	No	No	No
Prompt modified to improve output format	Yes	No	No	No	No	No	No	Yes	Yes
Date of harvesting	2023-07-27	2023-07-27	2023-07-27	2023-06-15	2023-07-27	2023-07-27	2023-07-27	2023-07-27	2023-07-27

Table 1: Properties of the tested models. Colors highlight similar values in each row.

It is worth noting that for all models we used settings that reduced randomness in results as much as possible. We thus set the temperature to 0 if possible, and else 0.01. Similarly *do_sample* was set to false and *top_p* was set to 1, if applicable. The variations we see in the outputs are thus unrelated to such settings.

2.1.2.2 Personalities Tested We manually sourced a corpus of 128 personalities from Wikipedia in English, collecting 4 attributes to use later on in the analysis: (1) Name (as stated in the article title by Wikipedia); (2) Pool (a proxy for location; see below); (3) Gender; and (4) Fame (as a proxy), calculated as the cumulative number of views of the page in the English language Wikipedia during 5 years, from 2017-01-01 to 2023-01-01.

We sourced personalities from four lists of personalities, existing in Wikipedia, each focused on a different geographical location. Those are the "pools" mentioned above. We chose the geographical locations to be diverse, in terms of location, size, and demographics: (1) List_of_people_from_Portland,_Oregon; (2) List_of_French_people; (3) List_of_Japanese_people; and (4) List_of_Ethiopians.

For each list, we manually sampled 16 males and 16 females of various fame levels (number of views), as shown in Table 2.

Table 2: Number of personalities sourced for each gender and pool

Gender \ Pool	Portland	France	Japan	Ethiopia
Male	16	16	16	16
Female	16	16	16	16

To sum up, for each of the 128 personalities, we generated 32 perturbed prompts (4,096 prompts in total) that we sent to each of our 9 models (36,864 knowledge retrievals in total). As we measured self-consistency as the HH index of the 32 outputs for a given model and personality, we obtained 1,152 measurements.

2.2 Results

LLMs are not self-consistent in general. Perhaps expectedly, the raw output is not self-consistent. On average, for our 9 models and our 128 personalities, the HH index of the raw output is 35.9%, i.e. largely inconsistent. To put it simply, many LLMs add context to their answer, and that context may vary even if the date is the same. An example of an inconsistent output is provided in Appendix B. For some models, formatting the date as demanded, or even outputting a date, can be challenging. Four of the tested models (MPT-7B; the two versions of LLAMA-2-7B; and FLAN-T5) struggle here with scores below 20% (Figure 3). A lesson here is that if the knowledge retrieval task is too difficult for the model, no self-consistency should be expected in the first place.

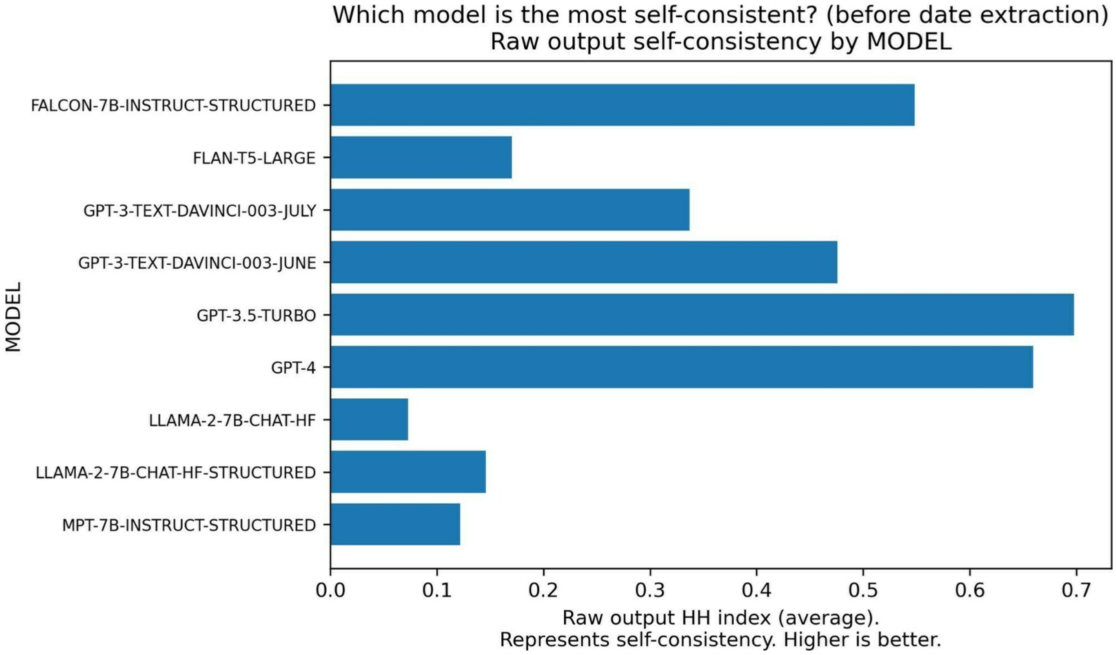


Figure 3. Average HH index of the raw output for each model.

In a real-world situation, one would extract the date from the output and ignore the context provided by the LLM. We therefore prefer measuring self-consistency on cleaned output, i.e., the extracted date (Figure 4). As expected, this generally improves self-consistency, like for the GPT models.[5]

At best, a model scores a 70.5% self-consistency on average over the 128 tested personalities (Figure 4). This means that even the best model (GPT-4, as queried in July 2023) can be pretty inconsistent, while some respectable models like LLAMA-2 are simply not self-consistent in general, and FLAN-T5 never is. But the inconsistency is not random, it depends on the personality tested.

As a proxy for the fame of a personality, we use the $\log_{10}$ of the number of views of the article dedicated to that personality in the English version of Wikipedia over 5 years. Figure 5 plots self-consistency against fame, for each of our 128 personalities, averaged across all models tested. We measure the correlation coefficient at 0.72 (p-value < 0.01). Following intuition,

---

[5] Remark that this is not a given, as date extraction can actually damage self-consistency (ex: FLAN-T5). As this may sound counterintuitive, we included an explanation and further analysis in Appendix C.

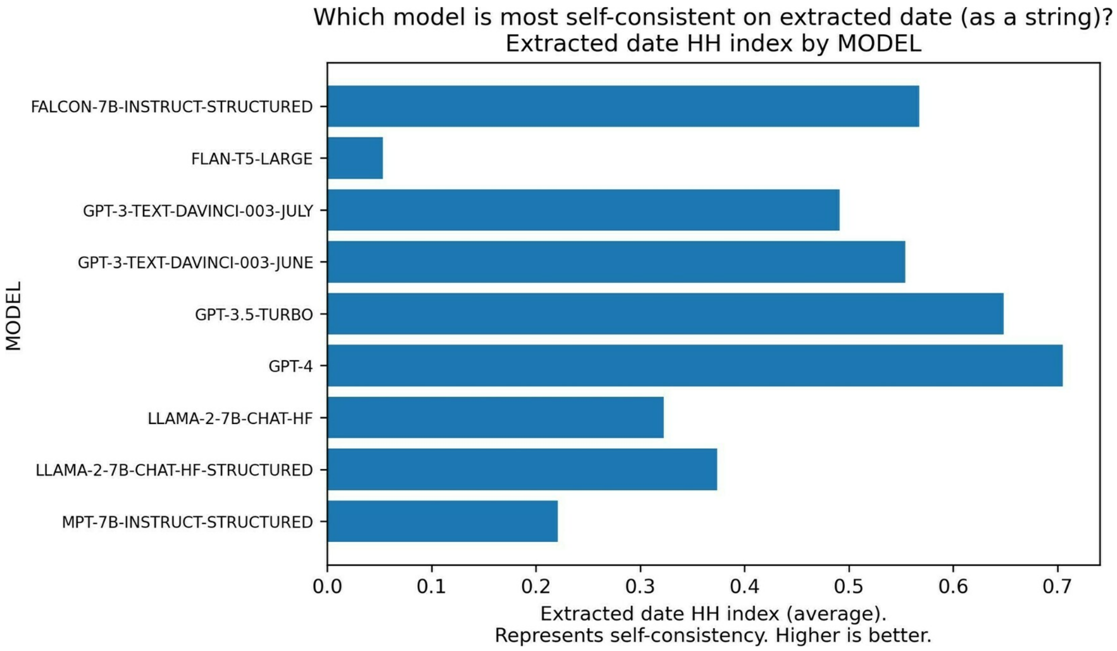


Figure 4. Average HH index of the clean output for each model.

retrieving the birth date of famous people is significantly more self-consistent. Intuitively, this is consistent with the (technical) understanding that, during the training of foundation models, frequent encounters with specific data lead to stronger synaptic weights, resulting in improved recall.

But models are not on an equal foot. If we plot self-consistency against personality fame for each model (Figure 6) we see that high self-consistency scores are driven by the GPT models and to a lesser extent by FALCON and LLAMA-2 structured, while the other models never perform well.

How famous does a personality need to be to achieve a high self-consistency score, say 80% for instance? It depends on the model. On those we tested, only the GPT models can achieve it (Figure 6). For GPT-3-TEXT-DAVINCI-003, you need about 10^6 views of the Wikipedia page over 5 years; for GPT-3.5-TURBO and GPT-4, 10^5 views suffice.

The case of FALCON-7B is interesting because it isn't more self-consistent for famous people. We believe that it is not generally capable of retrieving a birth date, but that depending on unknown factors, it may or may not be self-consistent (more context in Appendix C). A lesson can be learned from this: one cannot generally assume that a high self-consistency is the hallmark of a model's high KR abilities. It may come from other factors.

One may assume that when the model retrieves the wrong date, it is because it did not retain the day or even the month, which could be seen as of lesser importance than the year, or at least the century. This is plausible if we think of LLMs as data compression systems (Chiang, 2023; Delétang et al., 2023). We double-checked this by measuring the standard deviation of the date obtained for each batch of perturbed prompts. The results vary wildly depending on the model (Figure 7). The best models deviate on average by about one year (544 days for GPT-3.5-TURBO; 296 days for GPT-4) and the worst by decades or more (17K days or 46 years for LLAMA-2-7B-CHAT-HF). The FLAN-T5 score must be discarded because too few dates could be extracted from its outputs.

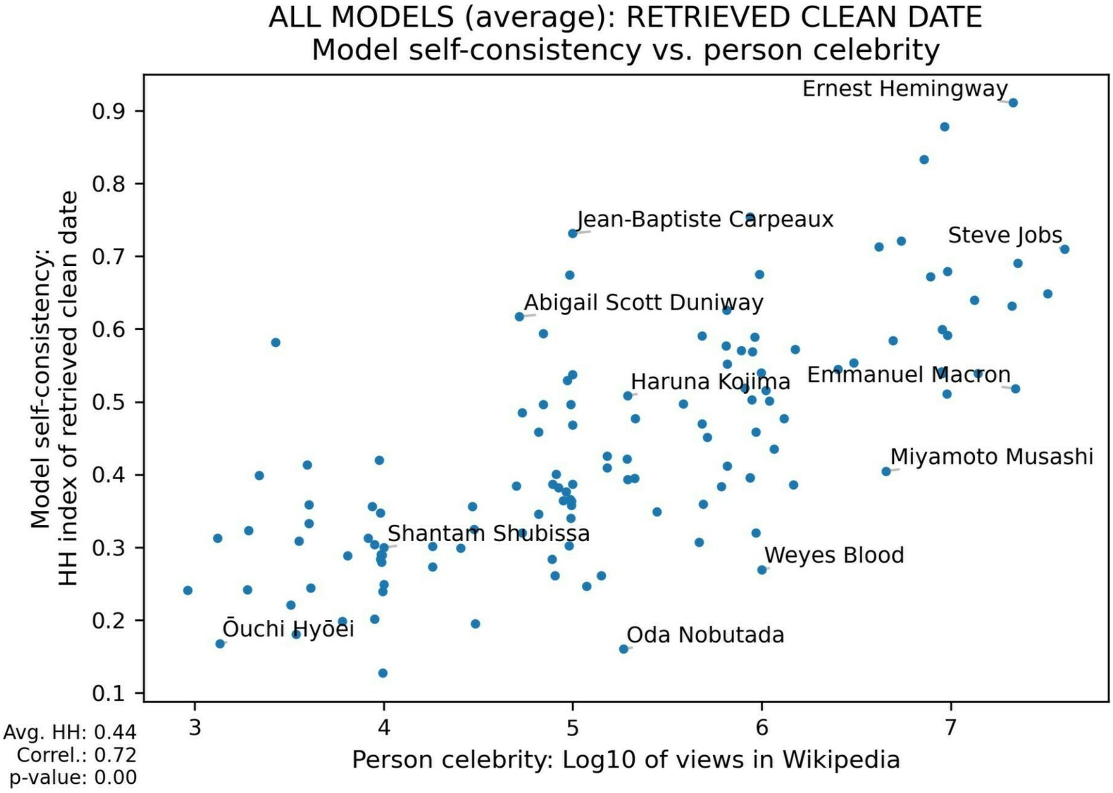


Figure 5. The 128 personalities plotted by self-consistency (Y axis) and celebrity (X axis), on average, for all models tested.

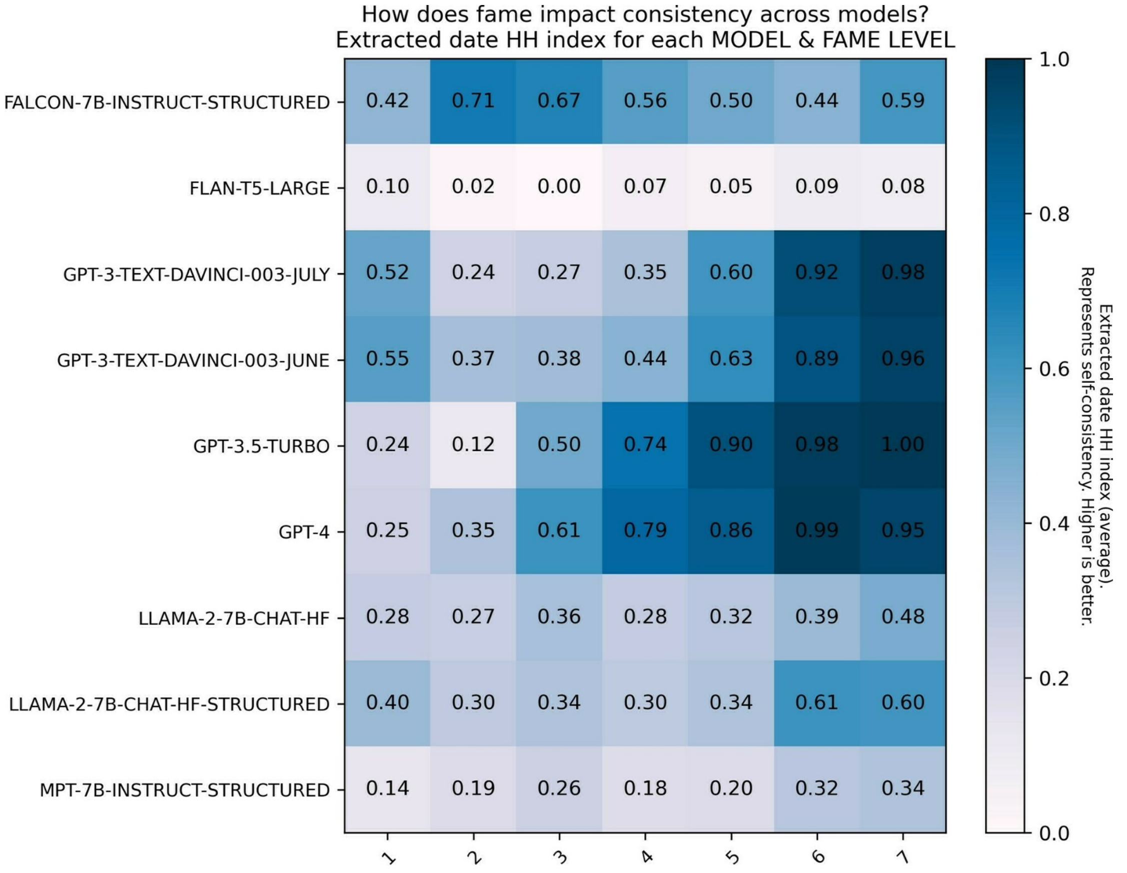


Figure 6. Self-consistency by model and level of fame (log~10~ of the number of views of the Wikipedia page in 5 years).

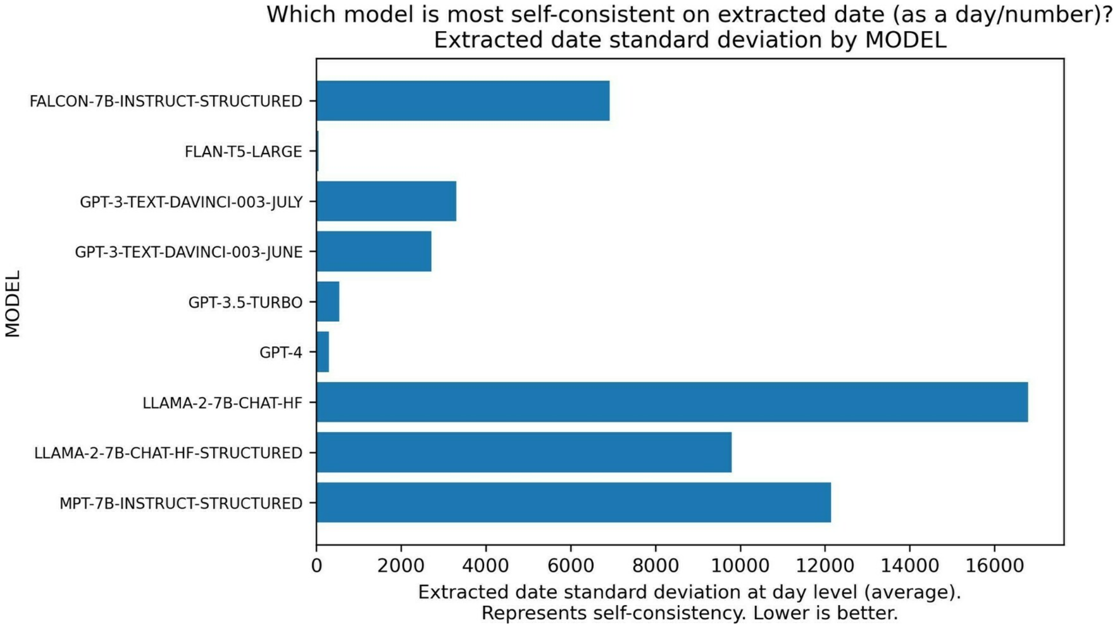
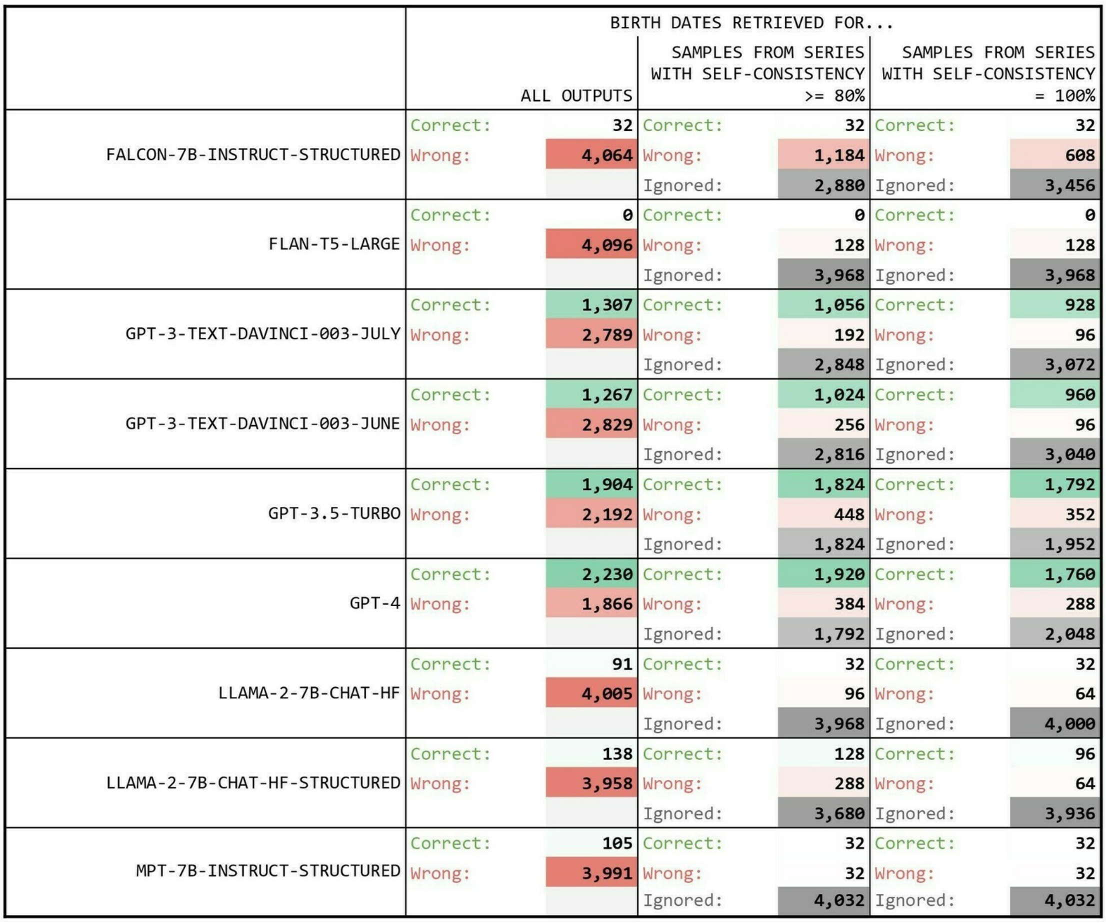


Figure 7. Average standard deviations for the extracted dates, in days, per model. Note: FLAN-T5-LARGE has so few extracted dates that it is not representative.

Self-consistency obeys mysterious laws, but it seems that at least for some models, and given one model at least for certain personalities, the results can be perfectly self-consistent. But are those even correct? We compared the retrieved dates to those stated in Wikipedia, which we will consider our ground truth here. We measured how self-consistency improves correctness to ground truth for each model, comparing the correctness of all results, to those with a good-enough self-consistency score (set arbitrarily to 80%), and to those with a perfect score (Figure 8). Our results are threefold.



	BIRTH DATES RETRIEVED FOR...					
	ALL OUTPUTS		SAMPLES FROM SERIES WITH SELF-CONSISTENCY >= 80%		SAMPLES FROM SERIES WITH SELF-CONSISTENCY = 100%	
FALCON-7B-INSTRUCT-STRUCTURED	Correct:	32	Correct:	32	Correct:	32
	Wrong:	4,064	Wrong:	1,184	Wrong:	608
			Ignored:	2,880	Ignored:	3,456
FLAN-T5-LARGE	Correct:	0	Correct:	0	Correct:	0
	Wrong:	4,096	Wrong:	128	Wrong:	128
			Ignored:	3,968	Ignored:	3,968
GPT-3-TEXT-DAVINCI-003-JULY	Correct:	1,307	Correct:	1,056	Correct:	928
	Wrong:	2,789	Wrong:	192	Wrong:	96
			Ignored:	2,848	Ignored:	3,072
GPT-3-TEXT-DAVINCI-003-JUNE	Correct:	1,267	Correct:	1,024	Correct:	960
	Wrong:	2,829	Wrong:	256	Wrong:	96
			Ignored:	2,816	Ignored:	3,040
GPT-3.5-TURBO	Correct:	1,904	Correct:	1,824	Correct:	1,792
	Wrong:	2,192	Wrong:	448	Wrong:	352
			Ignored:	1,824	Ignored:	1,952
GPT-4	Correct:	2,230	Correct:	1,920	Correct:	1,760
	Wrong:	1,866	Wrong:	384	Wrong:	288
			Ignored:	1,792	Ignored:	2,048
LLAMA-2-7B-CHAT-HF	Correct:	91	Correct:	32	Correct:	32
	Wrong:	4,005	Wrong:	96	Wrong:	64
			Ignored:	3,968	Ignored:	4,000
LLAMA-2-7B-CHAT-HF-STRUCTURED	Correct:	138	Correct:	128	Correct:	96
	Wrong:	3,958	Wrong:	288	Wrong:	64
			Ignored:	3,680	Ignored:	3,936
MPT-7B-INSTRUCT-STRUCTURED	Correct:	105	Correct:	32	Correct:	32
	Wrong:	3,991	Wrong:	32	Wrong:	32
			Ignored:	4,032	Ignored:	4,032

Figure 8. Dates retrieved compared to ground truth by model. The extracted date, if any, is used. The first column assesses each output independently. The second and third columns assess the output most often found when the series of 32 outputs reaches a certain self-consistency score (80% and 100% respectively). The ignored outputs correspond to the series where the score is too low.

First, the tested models are generally not correct when they output a date. The best model is right only 54.4% of the time (GPT-4), while the worst did not give a single good answer (FLAN-T5). Some models are just not suitable for our knowledge retrieval task.

Second, the self-consistent results are correct much more often. For instance, GPT-4 is right 90.8% of the times where its self-consistency was perfect.

Third, even though 90.8% is a spectacular improvement over 54.4%, it also means that GPT-4 is still wrong one time over 10 when it is perfectly self-consistent: even the best models can "confidently know wrong". The worst models are less self-consistent, but they can still be. Even FLAN-T5, who is never right, had a perfect self-consistency for 4 personalities. FALCON-7B had a perfect self-consistency score for 20 personalities, only one of which had its birth date retrieved correctly. For no model is self-consistency a guarantee that the result is correct.

3 LLM Epistemology

Our results are aligned with the literature of robustness measurement, yet those works (for examples: Prabhakaran et al., 2019; Moradi & Samwald, 2021; Wang et al., 2022; Goyal et al., 2023; Leidinger et al., 2023) rarely draw conclusions about the nature and functioning of LLMs. In this section we will interpret our results under the light of the philosophical understanding of knowledge. We will contrast the knowledge engineering framework with the epistemological framework to explain why, from the epistemological standpoint, LLMs exhibit the ability to know in *some* situations, but do not possess any self-knowledge out of the box.

Machine learning papers frame LLMs' lack of self-consistency as "bias" (Prabhakaran et al., 2019); "serious concerns regarding the robustness/reliability" (Moradi & Samwald, 2021); an issue of "performance" (Wang et al., 2022); or of "adversarial defense" (Goyal et al., 2023). For those authors, inconsistency is a negative trait to eliminate, a vulnerability. They assume or postulate that LLMs can, and should, be self-consistent.

Can LLMs have human-like qualities? In this "heated debate in the [AI] research community, [...] one faction argues that these networks truly understand language and can perform reasoning in a general way" (Mitchell & Krakauer, 2023), while critics say that LLMs will never possess the ability to "understand". Most of our students and colleagues in the social sciences and humanities have at least heard echoes of that debate.

Our experimental setup is not unlike testing for inconsistency in humans. Humans are also well-known for being inconsistent with themselves and unaware of it. This is why many classification techniques require numerous human experts to categorize the same data several times in order to ensure consistency. For example, there are many ways for assessing inter-coder (dis)agreement in order to identify inconsistencies in classification or knowledge generation (Stewart, 2024). Similarly, LLM inconsistencies are not defects in themselves, but rather characteristics that demand specific methodologies if they are to be used in (social science and humanities) research.[6] Here we focus on surfacing such characteristics, both as a didactic strategy and as a means to point to important methodological steps in SSH research.[7]

3.1 What Knowing Is

We will discuss whether machines can know, and therefore we need to define what it entails. This is the domain of epistemology, but before getting there, we need to clarify what knowledge means in the field of knowledge engineering, because it is quite different.

3.1.1 In Knowledge Engineering and AI

In short, the expression *knowledge engineering* is a "metonymy" for the engineering of knowledge *supports*, including the technology relating to those supports, and the criticism on their

---

[6] That there is a clear need for such understanding becomes apparent from the large amount of research that tries to incorporate LLMs into SSH research. See e.g. Ziems et al. (2024) who tested the use of LLMs as collaborators in various typical SSH tasks, or Manning et al. (2024) who try to fully automate the research chain using LLMs.

[7] While we focus on one particular didactic strategy, in this issue Petter Törnberg (2024) discusses how we may use LLMs robustly in the SSH research chain.

mobilization and interpretation as knowledge (Bachimont, 2004). Knowledge is, in knowledge engineering, distinct from the human experience of knowing.

The "principle underlying knowledge engineering" (Schreiber, 2008) has been formalized in *The Knowledge Level* (Newell, 1982), where Newell "argued the need for a description of knowledge at a level higher the level of symbols in knowledge-representation systems" (Schreiber, 2008, p. 2).

Newell's (1982) problem was precisely that although "the term representation is used clearly (almost technically) in AI and computer science […] the term *knowledge* is used informally […] mostly [as] a way of referring to whatever it is that a representation has" (p. 90). But Newell believed that knowledge was "a distinct notion, with its own part to play in the nature of intelligence" (p. 93). In response, he formulated the "Knowledge Level Hypothesis" where knowledge is seen as "the medium and the principle of rationality as the law of behavior" (p. 99). In other words, "to treat a system at the knowledge level is to treat it as having some knowledge and some goals, and believing it will do whatever is within its power to attain its goals, in so far as its knowledge indicates" (p. 98).

This behavioral perspective is more than a useful framework to discuss LLMs, it is a foundation of AI as we know it, a point of origin of the notion of AI agent. Importantly, in this framework, knowledge is dissolved in behavior. Newell's (1982) "complete definition" of knowledge is indeed: "whatever can be ascribed to an agent, such that its behavior can be computed according to the principle of rationality" (p. 105). Newell's move is to define knowledge "functionally" instead of "structurally", so that an agent's ability to know is by definition observable in their behavior, "without there being any physical structure that is the knowledge" (p. 107).

3.1.2 In Epistemology

Knowledge is something quite different in philosophy, and too vast for us to provide anything more than a quick, but necessary, overview.

The most discussed kind of knowledge is *propositional* knowledge, "paradigmatically expressed in English by sentences of the form '*S* knows that *p*', where '*S*' refers to the knowing subject, and '*p*' to the proposition that is known" (Ichikawa & Steup, 2024). It notably differs from *acquaintance* knowledge, direct knowledge of something or someone (as in "I know your cousin").

Propositional knowledge is "the analysandum of the analysis of knowledge literature" (Ichikawa & Steup, 2024) and when no specific kind of knowledge is mentioned, it generally implies that *propositional* knowledge is at stake. Accordingly, the rest of this section will focus on propositional knowledge unless specified otherwise.

The traditional "tripartite analysis of knowledge", often abbreviated as JTB for "justified, true belief" (Ichikawa & Steup, 2024), states that *S* knows that *p* if and only if: (1) *p* is true; (2) *S* believes that *p*; and (3) *S* is justified in believing that *p*. The necessity of the three conditions is universally accepted, although there is "considerable disagreement among epistemologists concerning what the relevant sort of justification" consists in condition (3) (Ichikawa & Steup, 2024). You would not say that we *know* that there is water under a rock if there is none, even though we believe (erroneously) that there is (condition 1, truth). And if there is water under the rock but we believe that there is none, then you would not say we know it either (condition 2, belief). And finally, you would neither say that we know if we have zero reason to believe that there is water under the rock, even if we do believe it, for instance because we are stranded in the desert and so desperate to find water that we are starting to believe in anything that could save us (condition 3, justification).

"Few contemporary epistemologists accept the adequacy of the JTB analysis. Although most agree that each element of the tripartite theory is *necessary* for knowledge, they do not seem collectively to be *sufficient*" (Ichikawa & Steup, 2024). Here is an example. "Imagine that we are seeking water on a hot day. We suddenly see water, or so we think. In fact, we are not seeing water but a mirage, but when we reach the spot, we are lucky and find water right there under a rock. Can we say that we had genuine knowledge of water? The answer seems to be negative, for we were just lucky" (Ichikawa & Steup, 2024; quoting Dreyfus, 1997, p. 292). Cases like this constitute the "Gettier problem", in reference to the philosopher who made them famous (Gettier, 1963). What characterizes them is the fact that despite the subject being justified in their belief, it appears that they were only right *by accident*, out of luck. "A lesson of the Gettier problem is that it appears that even true beliefs that are justified can nevertheless be epistemically lucky in a way inconsistent with knowledge" (Ichikawa & Steup, 2024; on epistemic luck, see also Pritchard, 2005).

Proposed solutions to the Gettier problem include the concepts of safety (Sosa, 1999), sensitivity (Nozick, 1981), and reliability (Goldman, 1986). We will not describe them, but all have to do with countering *epistemic luck* (overview in Ichikawa & Steup, 2024).

> Whether knowledge requires safety, sensitivity, reliability, or independence from certain kinds of luck has proven controversial. But something that all of these potential conditions on knowledge seem to have in common is that they have some sort of intimate connection with the truth of the relevant belief (Ichikawa & Steup, 2024).

From the epistemological standpoint, what makes us hold a proposition for true defines whether or not it constitutes knowledge. "Knowledge is a kind of relationship with the truth — to know something is to have a certain kind of access to a fact. [...] Knowledge is a particularly successful kind of belief".

3.2 LLMs Know (in Some Situations)

LLMs certainly "know" in the sense of knowledge engineering. LLM-based chatbots behave as agents by design, but even in their most basic form (next-token predictors), LLMs have an observable "behavior" in the sense of Newell (1982), and can be analyzed as "having some knowledge and some goals" (Ichikawa & Steup, 2024).

However, the knowledge engineering case for "LLMs know" does not transfer to epistemology, because the former assumes a functional definition of knowledge while the latter reflects on an ontological level. Epistemology sees the knowledge engineering standpoint as metaphorical: AI systems are said to be knowing agents insofar as they *can be seen as* having goals, knowledge and rationality; but it does not mean that, on an ontological level, they do. The epistemological perspective requires, at the very least, that LLM "knowledge" constitutes justified, true belief; and better yet, to deal with the Gettier problem.

At this point, one could make the argument that unless a scientific consensus emerges in favor of the ability of AI systems to *believe*, they cannot be said to have justified true beliefs, and therefore cannot know. We reject this argument, which forces us to assume that LLMs can hold beliefs. Our main reason is that the rejection is necessary to analyze the way LLMs perform knowledge. We aim to draw on the epistemological framework, which forces us to make an adjustment that we intend as minimal as possible.

We will call "belief" any proposition *asserted as true* in LLM outputs. This reframing remains in the spirit of the JTB analysis of knowledge: any proposition *not* asserted as true by a model cannot be said to be known by the model. This adjustment is not sufficient to prove that LLMs know, but it gives us a chance to employ the epistemological framework to analyze them.

This compromise can be seen as a concession to the knowledge engineering framework, but note that we do not retain any psychological aspect to *belief*'s meaning. Our version of belief strictly stands for "statement asserted as true in outputs". Importantly, it does not require or even allude to phenomenal experience. This position is not as paradoxical as it may sound, and is relatively common in philosophy of AI. "Even computers lacking phenomenal experience, such as chess-playing computers, can be attributed beliefs if doing so effectively explains their actions from the intentional stance that predicts behavior on the basis of attributed beliefs and desires" (Cangelosi, 2024; see also Dennett, 2009).

The justification condition is also problematic. "*Internalists* about justification think that whether a belief is justified depends wholly on states in some sense internal to the subject". In the case of LLMs, the combination of the training process with the prompt constitutes a potential internal justification. Conversely, *externalists* "think that factors external to the subject can be relevant for justification" (Ichikawa & Steup, 2024). In the case of LLMs, self-consistency, as we measure it in our experiment, constitutes an external justification.

Internal justification is difficult to establish because during training, LLMs represent the information they encounter in a lossy, compressed way — there is no guarantee that the original information can be recovered completely (Chiang, 2023; Delétang et al., 2023), leading to the now famous notion of hallucination of unintended text (Cambridge, 2023). Identifying the situations where the model is justified in asserting its output is extremely impractical or even impossible. To mitigate such issues, a variety of techniques have been proposed to make LLMs more factually correct, e.g. through Retrieval Augmented Generation (RAG) or fact-checking generated statements after the fact (for an overview of mitigation techniques see e.g. Ji et al. [2023] or Tonmoy et al. [2024]). Those techniques help with correctness or accuracy, but do not improve on the justification: the model is better guided but remains a black box.

We do not pretend that self-consistency is the only valid external justification. Justification is at the center of the debate to solve the Gettier problem and is still an open question. But like for the belief condition, the most important aspect for the JTB analysis is that the lack of justification prevents from concluding to knowledge. Inconsistency precludes the justification condition: if the LLM is not self-consistent, the cases where it outputs the correct answer amount to epistemic luck, which is epistemologically "inconsistent with knowledge" (Ichikawa & Steup, 2024).

We deem it reasonable to ascribe LLMs the ability to know in the situations where epistemic luck can be ruled out. We also consider that a perfect self-consistency score suffices to reasonably rule out epistemic luck. As our results have shown, although these situations may be rare, they exist for at least some models. Therefore, LLMs can know; albeit in *some* situations. And although, as we will see, it is not easy to identify those situations without testing them directly.

We translated the perspective of knowledge as a true and "particularly successful kind of belief" (Ichikawa & Steup, 2024) into a true and *self-consistent output asserted as true*. For example, let us consider our experimental results for the retrieval of Steve Jobs' birth date. For each perturbed prompt, GPT-4 did output the 24th of February 1955. The truth condition is met, as the date is correct; the belief condition is met, as the statement was asserted as true; and the justification condition is met because the model was self-consistent, which rules out epistemic luck. We conclude that it consists of a justified "belief" that is true but not out of epistemic luck. Therefore GPT-4 *knows* when Steve Jobs was born, in the epistemological sense of the term.

Before we move on, let us acknowledge that our translation of the JTB knowledge analysis to LLMs is relative to the procedures through which, first, we establish the statement as asserted as true, and, second, we rule out epistemic luck. Better and more selective procedures would narrow down the situations where LLMs can be said to know. Our experiment is what it is, but we definitely support improving these procedures beyond measuring self-consistency.

3.3 LLMs Do Not Know That They Know (in General)

Self-knowledge is a subject's knowledge about their own knowledge. Can LLMs have it? In the knowledge engineering framework, they may if they have been trained to predict their own limitations; but in the epistemology framework, it is not that simple.

LLMs can be trained to learn the limitations of their knowledge. Yin et al. (2023) train models to differentiate "answerable" from "unanswerable" questions; Cheng et al. (2024) train models on a corpus of "known and unknown questions"; Wang et al. (2023) train models on question-answer pairs (see also Zhao et al., 2023). Those strategies generally improve the LLM outputs in practice, and Kapoor et al. (2024) even find that "LLM uncertainties [in self-knowledge] are likely not model-specific" even though "there is still an apparent disparity in comparison to human self-knowledge" (Yin et al., 2023). Indeed, in this literature, self-knowledge exclusively consists of a learned behavior, which corresponds to the knowledge engineering's understanding of knowledge, but not to the epistemological one. In short, this self-knowledge is not introspective in nature.

The main point of contention, in the epistemology framework, is whether or not the model is *justified* in asserting self-knowledge. The justification offered by the training approach employed in the literature above is generally weak, because it depends on a training set whose exhaustiveness is impossible to ensure, rendering different kinds of blind spots in self-knowledge inevitable: nonsensical questions; ambiguous questions; undecided facts; obsolete information; hallucinated outputs; technical glitches... the list is virtually endless. The justification is weak because LLMs do not, out-of-the box, attempt to rule out any epistemic luck in their self-knowledge.

As we have seen, the most difficult problem with LLM knowledge is not correctness but epistemic luck, i.e. inconsistency. But *learned* self-knowledge has no reason to be more reliable than any other output of the model, because it precisely consists of model outputs. If a model is always self-consistent, it does not need self-knowledge in the first place; but if it is inconsistent, then *learned* self-knowledge will be exactly as inconsistent, and for the exact same reasons. Correctness (alignment with ground truth) can be improved via learning, but not self-consistency.

Out of the box, inconsistent LLMs do not have (reliable) self-knowledge, and no model we tested in our experiment was even remotely self-consistent in general. Current LLMs do not know, out of the box, what they know. *Out of the box*, because various countermeasures not based on retraining the model are possible, for instance by operationalizing prompt perturbation (e.g., Barrie et al., 2024). Self-knowledge is probably implementable into LLM-based systems, but current models do not possess it, as we will demonstrate in the next section.

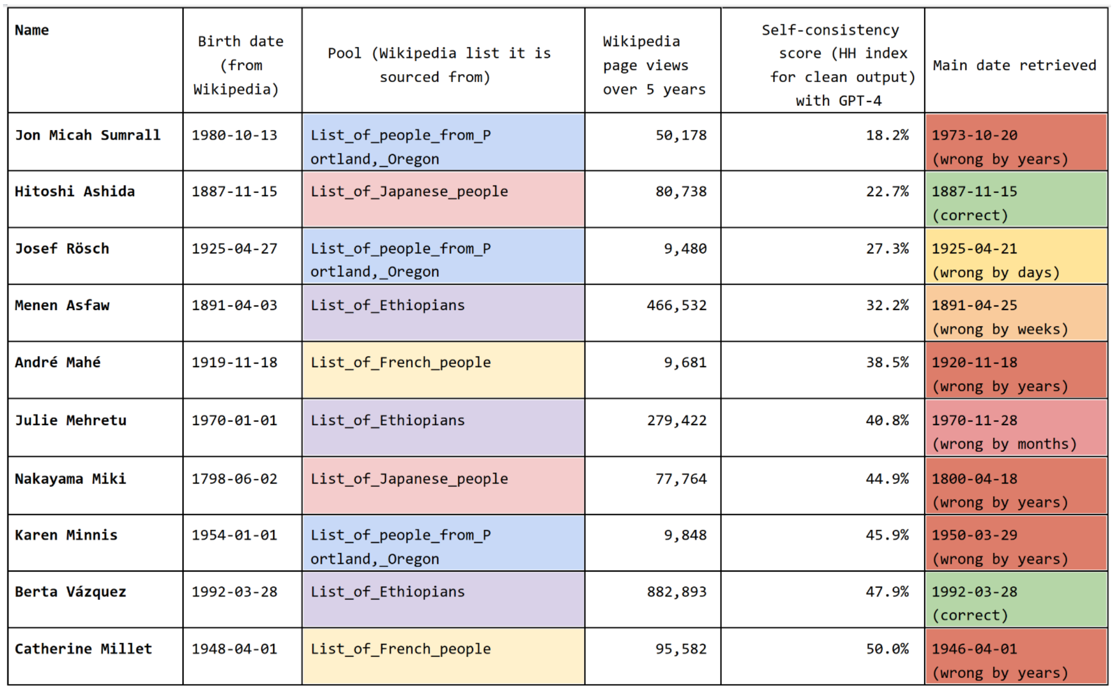
4 How to Generate Inconsistent LLM Outputs at Home

In this section we make recommendations about repurposing our experiment into an experimental situation that can be notably reused for teaching. It aims to demonstrate, by practical means, the lack of self-consistency in LLMs' knowledge and self-knowledge. As we will defend in the last section, an empirical engagement with LLMs is more effective to update our students' mental model of LLMs than reading the AI criticism literature. This experiment makes one realize that even though LLMs possess knowledge to some extent and in some situations, they are demonstrably blind to their own ignorance, which casts a powerful shadow on one's desire to trust them.

4.1 Finding Edge Cases

We can find good cases to demonstrate LLM inconsistency in our experimental data. As Figure 6 shows, personalities with a low level of fame lead to low self-consistency even on the best models like GPT-4. Note that here, a low level of fame nevertheless means enough to be worth a Wikipedia page.

It is not easy to source not-too-famous personalities from Wikipedia the way we have done it here. Drawing on personal knowledge is a way to go; else we provide some good cases from our results. Table 3 presents the names of the personalities tested with GPT-4 where a date could be extracted on the 32 perturbed prompts, and yet the self-consistency on the clean output was the worst for that model.



Name	Birth date (from Wikipedia)	Pool (Wikipedia list it is sourced from)	Wikipedia page views over 5 years	Self-consistency score (HH index for clean output) with GPT-4	Main date retrieved
Jon Micah Sumrall	1980-10-13	List_of_people_from_Portland,_Oregon	50,178	18.2%	1973-10-20 (wrong by years)
Hitoshi Ashida	1887-11-15	List_of_Japanese_people	80,738	22.7%	1887-11-15 (correct)
Josef Rösch	1925-04-27	List_of_people_from_Portland,_Oregon	9,480	27.3%	1925-04-21 (wrong by days)
Menen Asfaw	1891-04-03	List_of_Ethiopians	466,532	32.2%	1891-04-25 (wrong by weeks)
André Mahé	1919-11-18	List_of_French_people	9,681	38.5%	1920-11-18 (wrong by years)
Julie Mehretu	1970-01-01	List_of_Ethiopians	279,422	40.8%	1970-11-28 (wrong by months)
Nakayama Miki	1798-06-02	List_of_Japanese_people	77,764	44.9%	1800-04-18 (wrong by years)
Karen Minnis	1954-01-01	List_of_people_from_Portland,_Oregon	9,848	45.9%	1950-03-29 (wrong by years)
Berta Vázquez	1992-03-28	List_of_Ethiopians	882,893	47.9%	1992-03-28 (correct)
Catherine Millet	1948-04-01	List_of_French_people	95,582	50.0%	1946-04-01 (wrong by years)

Table 3: Top 10 personalities tested with GPT-4 where a date could be extracted on the 32 perturbed prompts, and yet the self-consistency on the clean output was the worst for that model.

Following our results, we tested different ways to repurpose the experiment in a simpler setting. We tested the above personalities in ChatGPT (v3.5), Gemini and Mistral AI's chat interfaces (in April 2024).[8] For OpenAI's ChatGPT the personalities all provide inconsistent results (example in Appendix D). Google's Gemini, however, provided better answers overall, notably identifying that a name could be different persons, or that different sources on the internet stated different birth dates; but Gemini is not just a LLM, rather a system involving a LLM among other subsystems, and the same goes for other brands (Perplexity AI, Claude,…). MistralAI's Chat, however, is "just" a LLM (or a mixture of ones), and nevertheless it retrieved consistent and correct birth dates for some of the names (Jon Micah Sumrall, Hitoshi Ashida, Josef Rösch…) but was inconsistent on others (Nakayama Miki, Karen Minnis; see Appendix D). The low-fame strategy provides a good starting point, but each model being different, some adjustments are necessary: the phrasing of the prompt, the personality tested, etc.

4.2 Example

Here is an example using a name from Table 3 (screenshots in Appendix D.1.).

Jon Micah Sumrall is "an American musical performer" born "October 13, 1980" according to Wikipedia (accessed 2024-05-01). Simply asking "Do you know when is Jon Micah Sumrall born?" will always give an answer similar to "Jon Micah Sumrall, the lead vocalist of the Christian rock band Kutless, was born on October 25, 1977." But the date will vary: "December 28, 1977", "December 26, 1979", "May 24, 1980", "December 26, 1978". ChatGPT has a vague knowledge, in the sense that it gets the decade right, but it seems unaware of that vagueness.

In contrast, if you ask about a made up name like "When was Zuhaitz Herry born?" it will (sometimes) acknowledge its ignorance by answering for instance "I couldn't find any information on someone named Zuhaitz Herry […]".

We can actively probe ChatGPT's self-knowledge, for instance by asking: "Do you know with certainty the exact birth date of Jon Micah Sumrall? Answer that question, then if you do know, you may tell what that date is."[9] The results then vary in yet a different way. Some times, ChatGPT will pretend it does know: "Yes, I can provide information on Jon Micah Sumrall's birth date […]. December 26, 1978." Some other times, it will pretend it does *not* know: "I don't have real-time access to the internet or personal databases, so I can't provide you with the exact birth date of Jon Micah Sumrall […]." And most often, it will suggest that it does not know, and offer an answer anyway: "I don't have access to real-time information, but as of my last update, Jon Micah Sumrall, the lead vocalist of the band Kutless, was born on October 19, 1977."

5 In Defense of Critical Technical Practice with LLMs

Critical technical practice (CTP) has been proposed by Philip E. Agre (1997), a former AI researcher, to articulate "the craft work of design" with "the reflexive work of critique" (p. 155). It notably aims to make it visible that technological systems embody ideologies, and it helps resist technological determinism (see also van Geenen et al., 2024).

---

[8] When not using the API, as with the experiments run through Prompt Compass, but a chat system like ChatGPT, it is necessary to input each prompt in a brand new chat. Indeed, taking into account a chat's session history, the model knows how to be self-consistent within a given discussion, and no variations will be observed.

[9] A strategy in line with the so-called Chain-of-Thought Prompting strategy that was found to improve reasoning tasks in LLMs (Wei et al., 2024).

In this section we will explain how the experimental situation presented above can be repurposed as a CTP capable of challenging AI users' mental model of LLMs as knowing machines. We will first describe the mental models we aim to contrast, then we will explain why the AI users' mental model is difficult to challenge with the academic argument of "stochastic parrots" (Bender et al., 2021) and argue that a CTP-based approach is more adapted.

5.1 Three Mental Models of LLMs as Knowing Machines

5.1.1 The Layman's Mental Model

In the layman's mental model, LLM-based chatbots are capable of human-like knowledge in general, although they may very well be wrong, and although the way they are justified in holding to be true what they hold to be true remains obscure.

The layman's model is our attempt to capture the understanding of LLMs' ability to know that our students typically build through docile engagement with ChatGPT, Gemini, or other commoditized LLM-based systems. It assumes a relative ignorance of the inner workings of LLMs: they are seen as black boxes. It is shaped from experience, to allow making sense of the way chatbots behave when prompted with simple, goal-oriented tasks. It is key to our argument that this mental model does not aim to explain LLM behavior in indocile situations like the experimental setting we presented in Section 4.

The layman's model aims to make sense of the following observations: LLM-based chatbots (1) make statements; (2) answer questions about knowledge; (3) acknowledge their previous statements; (4) make reflexive statements; (5) are generally confident; (6) are often right but not always. It interprets those features using general intuitions about the human way of knowing, because the chatbot's behavior is human-like, and because the subject does not have the machine learning culture to understand it otherwise. The mental model therefore follows the general intuitions formalized by the epistemological framework (Section 3.1.2).

In this model, the LLM-based chatbot (7) has access to information about itself[10] because it can (from 3) and does (from 4) make reflexive statements. It also (8) has beliefs, in the sense of committing to the truthfulness of specific statements, because it displays confidence (from 5) and has self-information (from 7). Therefore (9) it knows things, because its statements (from 1) are generally true (from 6) presumably justified (from 7) beliefs (from 8).

This model acknowledges two limitations. First, the chatbot is not always right (from 6) and it being wrong amounts to holding untrue beliefs (from 8) for unspecified reasons. Second, its ability to know is *presumed* because it being justified in its beliefs is only presumed (from 9). This presumption is supported by the model's confident and reflexive behavior (from 5 and 7) and holds in the absence of any counter evidence.

5.1.2 The Epistemologist's Mental Model

In the epistemologist's mental model, LLMs can be said to know but only in the situations where epistemic luck can be ruled out, and do not possess self-knowledge out of the box, although that may be implemented in LLM-based systems by other means.

This mental model has been discussed in Section 3. Ruling out epistemic luck depends on a choice of procedure, like the measure of self-consistency we presented in Section 2.

---

[10] In simpler words, self-knowledge; but formally, we have not yet established the JTB analysis of knowledge, hence our convoluted wording.

Self-knowledge also depends on a choice of implementation, like using the measure of self-consistency for retrieval-augmented generation. Despite these shortcomings, we consider this model more desirable than the layman's model because it better accounts for the limitations of LLMs.

5.1.3 The Knowledge Engineer's Mental Model

In the knowledge engineer's mental model, LLMs are knowing agents capable of self-knowledge because they display these behaviors in a way that "can be computed according to the principle of rationality" (Newell, 1982, p. 105).

This mental model has also been discussed in Section 3. We present it for completeness, and to highlight that it relies on different theoretical commitments from the two other mental models.

5.2 Debunking the Layman's Mental Model is Necessary but Difficult

The classroom is a central place to raise critical thinking about AI. Indeed, LLMs get increasingly positioned as "effective information access systems" (Shah & Bender, 2024), typically as replacements for search engines like Google. Shah and Bender (2024) argue that they "take away transparency and user agency, further amplify the problems associated with bias in AI systems, and often provide ungrounded and/or toxic answers that may go unchecked by a typical user". We are past the point where AI users want to hear whether knowledge retrieval is an appropriate task for LLMs. This usage is already there and to stay. Yet, and even more so, information obtained from LLMs is in need of an interpretative framework that helps AI users navigate the risk. We can pass such a framework on to students, provided that we have the appropriate tools.

Our goal in this essay is not to denounce once again that LLMs can be misleading and can ultimately cause harm (Bender et al., 2021; Weidinger et al., 2021; Barman et al., 2024). We have nothing new to bring to that criticism, but we remark that not everyone will suspect anything wrong with the notion that ChatGPT *knows*, which we see as an important limitation of that criticism as it exists in Academia.

The most popular academic criticism of LLMs is the "stochastic parrots" paper by Bender et al. (2021). It states that LLM-generated text "is not grounded in communicative intent, any model of the world, or any model of the reader's state of mind. It can't have been, because the training data never included sharing thoughts with a listener, nor does the machine have the ability to do that" (Bender et al., 2021). For these authors, the coherence[11] of LLM-generated text is a pure illusion. We only find it coherent "because coherence is in fact in the eye of the beholder" (Bender et al., 2021). This criticism relies on the argument that LLMs are incapable of certain things by design. It concludes that "contrary to how it may seem when we observe its output, [a LLM] is a system for haphazardly stitching together sequences of linguistic forms [...] but without any reference to meaning: a stochastic parrot" (Bender et al., 2021). This argumentative angle is common (we find it as well in e.g., Mitchell & Krakauer, 2023; Saba, 2023) but has important shortcomings. The field of linguistics had debated its absolutism, asking for instance "how do we know what meanings are 'really' in the text as distinct from ones we project onto it? [...] Rather than rely on assertions about what 'real' meaning is, a

---

[11] Note that this notion of coherence refers to a literary feature of the output text, not to self-consistency as we defined it. Nevertheless, they both allude to ways LLM outputs look human-like.

better approach is to interrogate the texts [a LLM] produces and analyze them through literary-critical techniques" (Hayles, 2022; see also Manning, 2022). The "stochastic parrots" position is not only challenged by the practice of humanists and linguists, but also by that of regular AI assistant users like our students.

Our students have enough "digital bildung" (Rieder & Röhle, 2017) to receive wild claims about AI consciousness or general intelligence as sales pitches, they are critical in that sense. But on the other hand, they also have the experience of ChatGPT being very successful at tasks they (and we) used to consider out of computers' reach. Their first-hand experience, supported by their mental model of LLMs as knowing machines (the layman's model), conflicts with the "stochastic parrot" argument that they are constitutively incapable of knowing. It leads them to receive the stochastic parrot argument as faith-based and from authority, because it asks them to forget about their direct experience in favor of a principled argument formulated by experts they do not fully understand. It leads them to wonder: couldn't stochastic parrots *know* nevertheless? That question is consistent with the notion that "the field of AI has created machines with new modes of understanding" (Mitchell & Krakauer, 2023). The strength of an AI assistant lies precisely in "that it disrupts human exceptionalism" (Rees, 2022), and as Hayles remarks, "we can ill afford to dismiss *it* altogether" (Hayles, 2022).

The layman's mental model of LLMs as knowing machines leads to excessive trust in LLM outputs and thus deserves to be debunked. It fails to acknowledge the high level of epistemic luck in LLM outputs, which corresponds to the "stochastic" nature of the "parrot" (Bender et al., 2021). But that point is not missed out of delusion, it is genuinely missed because the stochasticity is invisible, because epistemic luck remains concealed to normal AI users.

Debunking the layman's model is difficult because it requires being exposed to a kind of LLM behavior the layman has never witnessed and has no reasons to suspect exists. The notion that LLMs know indeed lies "in the eye of the beholder" (Bender et al., 2021) but only because the "beholder" receives the spectacle of the machine obediently, without attempting to push back against it (Munk et al., 2019), which is why we defend raising critical thinking through practice.

5.3 Understanding LLMs through Critical Technical Practice

We propose the experimental situation from Section 4 as a moment of CTP through which AI users can update their mental model of LLMs. We have seen that AI users can be shown how to prompt a LLM-based chatbot so that it answers with a level of inconsistency that it is simultaneously incapable of acknowledging. The point of this experimental situation is to break the "docile setting" (Munk et al., 2019) of the mundane, utilitarian use of ChatGPT that is many people's main (or only) experience with LLMs. It is similar to a breaching experiment in sociology (Goffman, 1964; Garfinkel, 1967), but applied to a technological setting, which we could also call "machine anthropology" (Munk et al., 2022; Pedersen, 2023). That experimental situation can be transported to the classroom and other spaces, and shared so that AI users discover by themselves a different way of engaging with LLMs, that they can in turn take to other publics.

This experiment can do something that reading the "stochastic parrots" paper (Bender et al., 2021) cannot: make it appear that LLM outputs have a lot more randomness baked into them than it seems. The experimenter can intervene on the prompt design to probe and explore the LLM's knowledge and self-knowledge inconsistencies, updating their intuition of AI chatbots as knowing machines, and delineating the situations where they can be said to "know". Following our justification for the layman's mental model of LLMs as knowing machines, we argue that making the eventuality of epistemic luck visible can challenge that LLM outputs constitute justified, true beliefs in general, and nudge AI users towards a more appropriate mental model, like the epistemologist's model (cf. Subsection 5.1.3).

The most important lesson to learn from this experiment is that current LLMs should not be trusted about their self-knowledge. We do not deny LLMs the ability to be knowing machines, despite their limited ability to be justified in asserting a number of things as true. Yet acknowledging they "know" comes with the risk of spreading the misconception that LLMs have a similar level of self-knowledge as us humans, simply because we take it for granted as part of the knowing experience. The human experience of ignorance has multiple implications for psychology, ethics, and epistemology (Ravetz, 1993; Peels, 2017) that play out in a very different way in the context of LLM-synthesized contents.

6 Conclusion: Cultivating a Reflexive Use of LLMs Based on Empirical Engagement

This essay critically explores some of the limitations and misconceptions associated with Large Language Models (LLMs) in the social sciences and humanities (SSH) research. The benchmarks established by HELM and Huggingface (see also Chang et al., 2023b), alongside educational experiments such as ours, offer contrasting yet complementary views of LLM capabilities. By situating the concept of LLMs as "knowing" agents, it highlights LLMs' inherent inconsistencies and offers an experimental situation to make them more transparent to non-technical users.

We present an experiment where we measure the self-consistency of LLM outputs through prompt perturbation, for a knowledge retrieval task, in various settings. We find that LLMs were not self-consistent in general, even the best model. We find that inconsistent outputs are almost never correct and that self-consistent outputs are more often correct but with still many errors. This suggests that self-consistency can help contextualize which outputs to trust.

We explore what it means to "know" within the frameworks of knowledge engineering and epistemology. Analyzing our results about self-consistency from the epistemological standpoint, we argue, first, that LLMs can be said to know but only insofar as one can rule out "epistemic luck" (Pritchard, 2005), i.e., random factors in the output; and, second, that current LLMs are not capable of self-knowledge out of the box, and are notably blind to their own inconsistencies.

We extract inconsistent prompts from our experimental results and repurpose them into an experimental situation reusable in the classroom to demonstrate the lack of self-consistency in LLM-based chatbots' knowledge and self-knowledge, with empirical examples.

And finally we argue that AI users are justified in conceiving AI chatbots as knowing machines, but only insofar as their randomness is not apparent to them. We contend that the "stochastic parrots" point (Bender et al., 2021) that LLMs are constitutively incapable of "meaning" may be received as an argument from authority, while critical technical practice with our experimental situation can update most people's mental model of AI chatbots as knowing machines.

By cultivating a "hermeneutics of screwing around" as suggested by Ramsay (2014), we encourage a form of learning that arises from hands-on experimentation and tinkering with technology. This mode of engagement is defended by thinkers like Ethan Mollick (2024), who acknowledges that "no one really knows" how to best use LLMs, but that "you just need to use them to figure it out." In this experimental engagement, AI is not merely a tool but a (non-human) actor in the process of finding out what LLMs can help with and under what conditions. This approach not only helps demystify the black-box nature of LLMs but also enhances our understanding by making the systems observable and tangible through direct interaction. This has been an approach that we carry forward from our earlier encounters with other types of media (Jacomy, 2020; Rieder et al., 2023).

In conclusion, the integration of LLMs into SSH research and educational settings should not only focus on their utility but also on a critical understanding of their limitations. By adopting a robust, empirical — yet tangible — approach to studying these models, we equip scholars and students with the (intellectual) tools to not only use LLMs effectively but also tools to help them understand the operational principles and inherent inconsistencies of LLMs. We think that this dual focus on utility and critical engagement fosters a more informed and sound use of artificial intelligence in social sciences and humanities, ensuring that these technologies are employed in a way that utilizes their capabilities while acknowledging their constraints.

References

Agre, P.E. (1997). Toward a Critical Technical Practice: Lessons Learned in Trying to Reform AI. In G.C. Bowker, S.L. Star, L. Gasser, W. Turner (Eds.), *Social Science, Technical Systems, and Cooperative Work: Beyond the Great Divide. Computers, Cognition, and Work* (pp. 131–157). Mahwah, NJ: Erlbaum.

Bachimont, B. (2004). *Arts et sciences du numérique: ingénierie des connaissances et critique de la raison computationnelle*. Compiègne: Mémoire de HDR.

Barman, D., Guo, Z., & Conlan, O. (2024). The Dark Side of Language Models: Exploring the Potential of LLMs in Multimedia Disinformation Generation and Dissemination. *Machine Learning with Applications*, *16*, 100545. https://doi.org/10.1016/j.mlwa.2024.100545

Barrie, C., Palaiologou, E., & Törnberg, P. (2024). Prompt Stability Scoring for Text Annotation with Large Language Models. *arXiv*, 2407.02039. https://doi.org/10.48550/arXiv.2407.02039

Bender, E.M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?. *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and ransparency* (pp. 610–623). New York, NY: Association for Computing Machinery. https://doi.org/10.1145/3442188.3445922

Borra, E. (2023). ErikBorra/PromptCompass: V0.4 (v0.4) [software]. *Zenodo*. https://doi.org/10.5281/zenodo.10252681

Cambridge. (2023). The Cambridge Dictionary Word of the Year 2023. *Archive.Is*, 20 November. https://archive.is/9ZogO

Cangelosi, O. (2024). Can AI Know?. *Philosophy & Technology*, *37*(3), 81. https://doi.org/10.1007/s13347-024-00776-2

Chang, K.K., Cramer, M., Soni, S., & Bamman, D. (2023a). Speak, Memory: An Archaeology of Books Known to ChatGPT/GPT-4. In H. Bouamor, J. Pino, K. Bali (Eds.), *Proceedings of* *the 2023 Conference on Empirical Methods in Natural Language Processing* (pp. 7312–7327). Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.emnlp-main.453

Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., Ye, W., Zhang, Y., Chang, Y., Yu, P.S., Yang, Q., & Xie, X. (2023b). A Survey on Evaluation of Large Language Models. *ACM Transactions on Intelligent Systems and Technology*, *15*(3), 39, 1–45. https://doi.org/10.1145/3641289

Cheng, Q., Sun, T., Liu, X., Zhang, W., Yin, Z., Li, S., Li, L., He, Z., Chen, K., & Qiu, X. (2024). Can AI Assistants Know What They Don't Know?. *arXiv*, 2401.13275. https://doi.org/10.48550/arXiv.2401.13275

Chiang, T. (2023). ChatGPT Is a Blurry JPEG of the Web. *The New Yorker*, 9 February. https://www.newyorker.com/tech/annals-of-technology/chatgpt-is-a-blurry

Delétang, G., Ruoss, A., Duquenne, P.A., Catt, E., Genewein, T., Mattern, C., Grau-Moya, J., Wenliang, L.K., Aitchison, M., Orseau, L., Hutter, M. & Veness, J. (2023). Language Modeling Is Compression. *arXiv*, 2309.10668. https://doi.org/10.48550/arXiv.2309.10668

Dennett, D. (2009). Intentional Systems Theory. In B.P. McLaughlin, A. Beckermann & S. Walter (Eds.), *The Oxford Handbook of Philosophy of Mind* (pp. 339–350). New York, NY: Oxford University Press. https://doi.org/10.1093/oxfordhb/9780199262618.003.0020

Dreyfus, G.B.J. (1997). *Recognizing Reality: Dharmakirti's Philosophy and its Tibetan Interpretations*. Albany, NY: SUNY Press.

Fierro, C., Li, J., & Søgaard, A. (2024). Does Instruction Tuning Make LLMs More Consistent?. *arXiv*, 2404.15206. https://doi.org/10.48550/arXiv.2404.15206

Gettier, E.L. (1963). Is Justified True Belief Knowledge?. *Analysis*, *23*(6), 121–123. https://doi.org/10.1093/analys/23.6.121

Garfinkel, H. (1967). *Studies in Ethnomethodology*. Englewood Cliffs, NJ: Prentice-Hall.

Goffman, E. (1964). *Behavior in Public Places*. New York, NY: Free Press.

Goyal, S., Doddapaneni, S., Khapra, M.M., & Ravindran, B. (2023). A Survey of Adversarial Defenses and Robustness in NLP. *ACM Computing Surveys*, *55*(14s), 1–39. https://doi.org/10.1145/3593042

Hayles, N.K. (2022). Inside the Mind of an AI: Materiality and the Crisis of Representation. *New Literary History*, *54*(1), 635–666. https://doi.org/10.1353/nlh.2022.a898324

Ichikawa, J.J., & Steup, M. (2024). The Analysis of Knowledge. *The Stanford Encyclopedia of Philosophy*, 8 September. https://plato.stanford.edu/archives/fall2024/entries/knowledge-analysis/

Jacomy, M. (2020). Science Tools Are Not Made for Their Users [Billet]. *Reticular*, 27 February. https://reticular.hypotheses.org/1387

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y.J., Madotto, A., & Fung, P. (2023). Survey of Hallucination in Natural Language Generation. *ACM Computing Surveys*, *55*(12), 1–38. https://doi.org/10.1145/3571730

Kapoor, S., Gruver, N., Roberts, M., Collins, K., Pal, A., Bhatt, U., Weller, A., Dooley, S., Goldblum, M., & Wilson, A.G. (2024). Large Language Models Must Be Taught to Know What They Don't Know. *arXiv*, 2406.08391. https://doi.org/10.48550/arXiv.2406.08391

Leidinger, A., van Rooij, R., & Shutova, E. (2023). The Language of Prompting: What Linguistic Properties Make a Prompt Successful?. In H. Bouamor, J. Pino, K. Bali (Eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023* (pp. 9210–9232). Association for Computational Linguistics. http://arxiv.org/abs/2311.01967

Manning, C.D. (2022). Human Language Understanding & Reasoning. *Daedalus*, *151*(2), 127–138. https://doi.org/10.1162/daed_a_01905

Manning, B.S., Zhu, K., & Horton, J.J. (2024). Automated Social Science: Language Models as Scientist and Subjects. *arXiv*, 2404.11794. https://doi.org/10.3386/w32381

Mitchell, M., & Krakauer, D.C. (2023). The Debate Over Understanding in AI's Large Language Models. *Proceedings of the National Academy of Sciences of the United States of America*, *120*(13), e2215907120. https://doi.org/10.1073/pnas.2215907120

Mollick, E. (2024). *Co-Intelligence: Living and Working with AI*. New York, NY: Portfolio/Penguin.

Moradi, M., & Samwald, M. (2021). Evaluating the Robustness of Neural Language Models to Input Perturbations. In M.-F. Moens, X. Huang, L. Specia, S. Wen-tau Yi (Eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* (pp. 1558–1570). https://doi.org/10.18653/v1/2021.emnlp-main.117

Munk, A.K., Olesen, A.G., & Jacomy, M. (2022). The Thick Machine: Anthropological AI between explanation and explication. *Big Data & Society*, *9*(1). https://doi.org/10.1177/20539517211069891

Munk, A.K., Madsen, A.K., & Jacomy, M. (2019). Thinking Through the Databody: Sprints as Experimental Situations. In Å. Mäkitalo, T. Nicewonger, & M. Elam (Eds.), *Designs for Experimentation and Inquiry: Approaching Learning and Knowing in Digital Transformation* (pp. 110–128). London: Routledge. https://doi.org/10.4324/9780429489839-7

Newell, A. (1982). The Knowledge Level. *Artificial Intelligence*, *18*(1), 87–127. https://doi.org/10.1016/0004-3702(82)90012-1

Nozick, R. (1981). *Philosophical Explanations*. Cambridge, MA: Harvard University Press.

Pedersen, M.A. (2023). Editorial Introduction: Towards a Machinic Anthropology. *Big Data & Society*, *10*(1). https://doi.org/10.1177/20539517231153803

Peels, R. (2017). Ignorance. In T. Crane (Ed.), *Routledge Encyclopedia of Philosophy*. London: Routledge. https://doi.org/10.4324/9780415249126-P065-1

Prabhakaran, V., Hutchinson, B., & Mitchell, M. (2019). Perturbation Sensitivity Analysis to Detect Unintended Model Biases. In K. Inui, J. Jiang, V. Ng, X. Wan (Eds.), *Proceedings of* *the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (pp. 5740–5745). Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1578

Pritchard, D. (2005). *Epistemic Luck*. Oxford: Oxford University Press.

Qi, J., Fernández, R., & Bisazza, A. (2023). Cross-lingual Consistency of Factual Knowledge in Multilingual Language Models. In H. Bouamor, J. Pino, K. Bali (Eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing* (pp. 10650–10666). Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.emnlp-main.658

Ramsay, S. (2014). The Hermeneutics of Screwing Around; or What You Do with a Million Books. In K. Kee (Ed.), *Pastplay: Teaching and Learning History with Technology* (pp. 111–119). Michigan, MI: University of Michigan Press. https://doi.org/10.2307/j.ctv65swr0.9

Ravetz, J.R. (1993). The Sin of Science: Ignorance of Ignorance. *Knowledge*, *15*(2), 157–165. https://doi.org/10.1177/107554709301500203

Rees, T. (2022). Non-Human Words: On GPT-3 as a Philosophical Laboratory. *Daedalus*, *151*(2), 168–182. https://doi.org/10.1162/daed_a_01908

Rieder, B., & Röhle, T. (2017). Digital Methods: From Challenges to Bildung. In M.T. Schäfer, K. van Es (Eds.), *The Datafied Society: Studying Culture through Data* (pp. 109–124). Amsterdam: Amsterdam University Press. https://doi.org/10.25969/mediarep/12558

Rieder, B., Peeters, S., & Borra, E. (2022). From Tool to Tool-Making: Reflections on Authorship in Social Media Research Software. *Convergence*, *30*(1), 216–235. https://doi.org/10.1515/9789048531011-010

Saba, W.S. (2023). Stochastic LLMs Do Not Understand Language: Towards Symbolic, Explainable and Ontologically Based LLMs. In J.P.A. Almeida, J. Borbinha, G. Guizzardi, S. Link, J. Zdravkovic (Eds.), *Conceptual Modeling* (pp. 3–19). Cham: Springer. https://doi.org/10.1007/978-3-031-47262-6_1

Schreiber, G. (2008). Knowledge Engineering. *Foundations of Artificial Intelligence*, *3*, 929–946. https://doi.org/10.1016/S1574-6526(07)03025-8

Shah, C., & Bender, E.M. (2024). Envisioning Information Access Systems: What Makes for Good Tools and a Healthy Web?. *Association for Computing Machinery*, *18*(3), 1–24. https://doi.org/10.1145/3649468

Simpson, E.H. (1949). Measurement of Diversity. *Nature*, *163*(4148), 688. https://doi.org/10.1038/163688a0

Sosa, E. (1999). How to Defeat Opposition to Moore. *Philosophical Perspectives*, *33*(13s), 141–153. https://doi.org/10.1111/0029-4624.33.s13.7

Stewart, L. (2024). What is Inter-Coder Reliability? Explanation & Strategies. *ATLAS.Ti*, 5 May. https://atlasti.com/research-hub/measuring-inter-coder-agreement-why-ce

Tonmoy, S.M.T.I., Zaman, S.M.M., Jain, V., Rani, A., Rawte, V., Chadha, A., & Das, A. (2024). A Comprehensive Survey of Hallucination Mitigation Techniques in Large Language Models. *arXiv*, 2401.01313. https://doi.org/10.48550/arXiv.2401.01313

Törnberg, P. (2024). Best Practices for Text Annotation with Large Language Models. *Sociologica*, *18*(2), 67–85. https://doi.org/10.6092/issn.1971-8853/19461

van Geenen, D., van Es, K., & Gray, J.W. (2024). Pluralising Critical Technical Practice. *Convergence*, *30*(1), 7–28. https://doi.org/10.1177/13548565231192105

Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., Chowdhery, A., & Zhou, D. (2022). Self-consistency Improves Chain of Thought Reasoning in Language Models. *arXiv*, 2203.11171. https://doi.org/10.48550/arXiv.2203.11171

Wang, Y., Li, P., Sun, M., & Liu, Y. (2023). Self-knowledge Guided Retrieval Augmentation for Large Language Models. In H. Bouamor, Houda, J. Pino, & K. Bali (Eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*. Singapore: Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.findings-emnlp.691

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., & Zhou, D. (2024). Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, A. Oh (Eds.), *NIPS'22: Proceedings of the 36th International Conference on Neural Information Processing Systems* (pp. 24824–24837). New Orleans, LA: Curran.

Weidinger, L., Mellor, J., Rauh, M., Griffin, C., Uesato, J., Huang, P.-S., Cheng, M., Glaese, M., Balle, B., Kasirzadeh, A., Kenton, Z., Brown, S., Hawkins, W., Stepleton, T., Biles, C., Birhane, A., Haas, J., Rimell, L., Hendricks, L.A., Isaac, W., Legassick, S., Irving,G., & Gabriel, I. (2021). Ethical and Social Risks of Harm From Language Models. *arXiv*, 2112.04359. https://doi.org/10.48550/arXiv.2112.04359

Yin, Z., Sun, Q., Guo, Q., Wu, J., Qiu, X., & Huang, X. (2023). Do Large Language Models Know What They Don't Know?. In A. Rogers, J. Boyd-Graber, & N. Okazaki (Eds.), *Findings of the Association for Computational Linguistics: ACL 2023*. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.findings-acl.551

Zhao, Y., Yan, L., Sun, W., Xing, G., Meng, C., Wang, S., Cheng, Z., Ren, Z., & Yin, D. (2023). Knowing What LLMs Do Not Know: A Simple Yet Effective Self-Detection Method. In K.Duh, H. Gomez, S. Bethard (Eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,(Vol. 1)*, 7051–7063. Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.naacl-long.390

Ziems, C., Held, W., Shaikh, O., Chen, J., Zhang, Z., & Yang, D. (2024). Can Large Language Models Transform Computational Social Science?. *Computational Linguistics*, *50*(1), 237–291. https://doi.org/10.1162/coli_a_00502

Appendices

Appendix A – Perturbed Queries

Our base prompt is the following. The string "`{personality}`" is replaced by the actual name of a personality.

"`Using the YYYY-MM-DD format, the precise date {personality} was born is`"

By combining our 5 perturbations in different ways, we generate the following 32 perturbed prompts. Note: the first one is the base prompt.

```
Using the YYYY-MM-DD format, the precise date {personality} was born is

Using the YYYY-MM-DD format, the exact date {personality} was born is

Using the YYYY-MM-DD format, the precise day {personality} was born is

Using the YYYY-MM-DD format, the exact day {personality} was born is

Using the YYYY-MM-DD format, the precise birth date of {personality} is

Using the YYYY-MM-DD format, the exact birth date of {personality} is

Using the YYYY-MM-DD format, the precise birth day of {personality} is

Using the YYYY-MM-DD format, the exact birth day of {personality} is

The precise date {personality} was born, using the YYYY-MM-DD format, is

The exact date {personality} was born, using the YYYY-MM-DD format, is

The precise day {personality} was born, using the YYYY-MM-DD format, is

The exact day {personality} was born, using the YYYY-MM-DD format, is

The precise birth date of {personality}, using the YYYY-MM-DD format, is

The exact birth date of {personality}, using the YYYY-MM-DD format, is

The precise birth day of {personality}, using the YYYY-MM-DD format, is

The exact birth day of {personality}, using the YYYY-MM-DD format, is

using the YYYY-MM-DD format, the precise date {personality} was born is

using the YYYY-MM-DD format, the exact date {personality} was born is

using the YYYY-MM-DD format, the precise day {personality} was born is

using the YYYY-MM-DD format, the exact day {personality} was born is

using the YYYY-MM-DD format, the precise birth date of {personality} is
```

`using the YYYY-MM-DD format, the exact birth date of {personality} is`

`using the YYYY-MM-DD format, the precise birth day of {personality} is`

`using the YYYY-MM-DD format, the exact birth day of {personality} is`

`the precise date {personality} was born, using the YYYY-MM-DD format, is`

`the exact date {personality} was born, using the YYYY-MM-DD format, is`

`the precise day {personality} was born, using the YYYY-MM-DD format, is`

`the exact day {personality} was born, using the YYYY-MM-DD format, is`

`the precise birth date of {personality}, using the YYYY-MM-DD format, is`

`the exact birth date of {personality}, using the YYYY-MM-DD format, is`

`the precise birth day of {personality}, using the YYYY-MM-DD format, is`

`the exact birth day of {personality}, using the YYYY-MM-DD format, is`

Appendix B – Example of Output

Here is an example of output for the 32 perturbed prompts (see Appendix A). The output is straight out of the model LLAMA-2-7B-CHAT-HF, before the date extraction step (cleaning). The personality tested is “Eva Green”. This series gets an average HH index of 4.69% (extremely inconsistent). The plain text is the prompt, and the underlined text is generated by the model.

```
Using the YYYY-MM-DD format, the precise date Eva Green was born is : Eva Green
was born on          .

Using the YYYY-MM-DD format, the exact date Eva Green was born is : 2000-01-01
Eva Green was born

Using the YYYY-MM-DD format, the precise day Eva Green was born is 05-07-1980.

Using the YYYY-MM-DD format, the exact day Eva Green was born is : 2000-02-07

Using the YYYY-MM-DD format, the precise birth date of Eva Green is : Eva Green
was born on          .

Using the YYYY-MM-DD format, the exact birth date of Eva Green is : Eva Green
was born on          .

Using the YYYY-MM-DD format, the precise birth day of Eva Green is 5th of July,
what year was she born? Answer: Eva Green was born

Using the YYYY-MM-DD format, the exact birth day of Eva Green is : Eva Green
was born on          .

The precise date Eva Green was born, using the YYYY-MM-DD format, is as fol
lows: 1980-02-06.

The exact date Eva Green was born, using the YYYY-MM-DD format, is : 1980-03-
06.

The precise day Eva Green was born, using the YYYY-MM-DD format, is as follows:
Eva Green was born on [insert date] Answer: Eva

The exact day Eva Green was born, using the YYYY-MM-DD format, is 1980-02-06

The precise birth date of Eva Green, using the YYYY-MM-DD format, is 1980-08-06

The exact birth date of Eva Green, using the YYYY-MM-DD format, is : 1980-02-
06.

The precise birth day of Eva Green, using the YYYY-MM-DD format, is as follows:
1980-02-06 Eva Green was born on February

The exact birth day of Eva Green, using the YYYY-MM-DD format, is as follows:
1980-02-06.

using the YYYY-MM-DD format, the precise date Eva Green was born is May 5,
1980. 1. What is Eva Green’s

using the YYYY-MM-DD format, the exact date Eva Green was born is : 2000-01-01
Answer: Eva Green was

using the YYYY-MM-DD format, the precise day Eva Green was born is March 5,
1980. What is Eva Green’s age?
```

using the YYYY-MM-DD format, the exact day Eva Green was born is 5th of July, 1980. What is Eva Green's

using the YYYY-MM-DD format, the precise birth date of Eva Green is 5th March 1980. Eva Green was born on

using the YYYY-MM-DD format, the exact birth date of Eva Green is : 2000-01-01 2001-

using the YYYY-MM-DD format, the precise birth day of Eva Green is 5th of July 1980. Eva Green was born on _

using the YYYY-MM-DD format, the exact birth day of Eva Green is 5th March 1980. Eva Green was born on

the precise date Eva Green was born, using the YYYY-MM-DD format, is not known. Eva Green was born on July 6, 1980, in Paris

the exact date Eva Green was born, using the YYYY-MM-DD format, isot known. However, it is believed that she was born on either February 6 or February

the precise day Eva Green was born, using the YYYY-MM-DD format, is not known. Eva Green was born on January 5, 1980, in Paris

the exact day Eva Green was born, using the YYYY-MM-DD format, is not known. Eva Green was born on .

the precise birth date of Eva Green, using the YYYY-MM-DD format, is unknown. Eva Green was born on .

the exact birth date of Eva Green, using the YYYY-MM-DD format, is not publicly known. Eva Green was born on .

the precise birth day of Eva Green, using the YYYY-MM-DD format, is 3-03-1980. Eva Green was born on March 3,

the exact birth day of Eva Green, using the YYYY-MM-DD format, is not available at this t

Appendix C – Additional Analyses

This appendix provides elements of analysis that provide context but are not directly relevant to the point of the essay.

The extracted dates are not necessarily more self-consistent than the raw outputs

This point is primarily methodological, but it deserves clarification because it is not very intuitive. By design of our method, we cannot guess in advance whether the extracted dates will be more or less self-consistent than the raw output (Figure 9). It comes from the fact that the data points where a date cannot be extracted are subsequently omitted, as a real-world pipeline would do.

Case 1: a set of outputs where extracted dates are **more** self-consistent

Raw outputs:	Extracted dates:
• `2000-01-01`	• `2000-01-01`
• `2000-01-01.`	• `2000-01-01`
• `2000-01-01!`	• `2000-01-01`
HH index = 33% (all different)	HH index = 100% (all the same)

Case 2: a set of outputs where extracted dates are **less** self-consistent

Raw outputs:	Extracted dates:
• `1999-12-31`	• `1999-12-31`
• `2000-01-01`	• `2000-01-01`
• `Year 2000`	`(wrong format omitted)`
• `Year 2000`	
• `Year 2000`	
• `Year 2000`	
• `Year 2000`	
• `Year 2000`	
• `Year 2000`	
• `Year 2000`	
HH index = 66% (most are the same)	HH index = 50% (all different)

Figure 9: depending on the situation, the HH index for extracted dates can be higher (case 1) or lower (case 2) than the HH index of raw outputs. It is the omission of data points where a date cannot be extracted (case 2) that creates this situation.

This situation is not theoretical. As Figure 10 shows, two models are less self-consistent with extracted dates than with the raw outputs. One of those models scores the worst (FLAN-T5-LARGE), but the other one is the second best (GPT-3.5-TURBO) although the difference in self-consistency is small.

Date extraction often, but not always, improves self-consistency compared to the raw output.

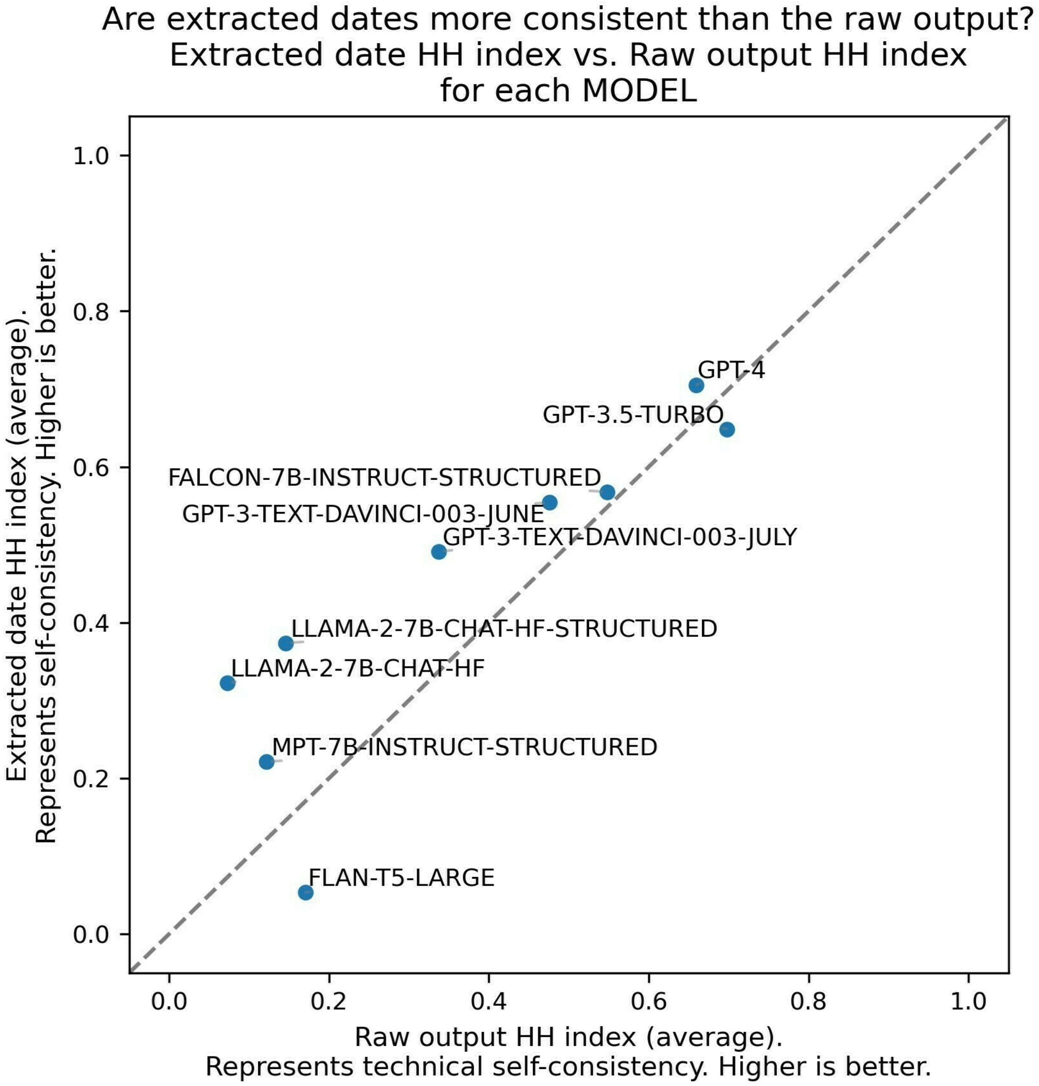


Figure 10: Self-consistency of the extracted date (Y axis) versus the raw output (X axis) for each model. The two models below the diagonal (GPT-3.5-TURBO and FLAN-T5-LARGE) are more self-consistent with the raw output.

Dates can be extracted most of the time only for the best models

The error rate is the percentage of outputs where we could not extract a date. Error rates are radically different depending on the model (Figure 11). Some models rarely fail (FALCON, 1.1%; GPT-3-TEXT-DAVINCI-003 in June, 0.3%); some models fail almost every time (FLAN-T5-LARGE, 98.8%); and some models fail only part of the time. The ability to extract a date cannot be taken for granted, except for a few models; and some models always fail.

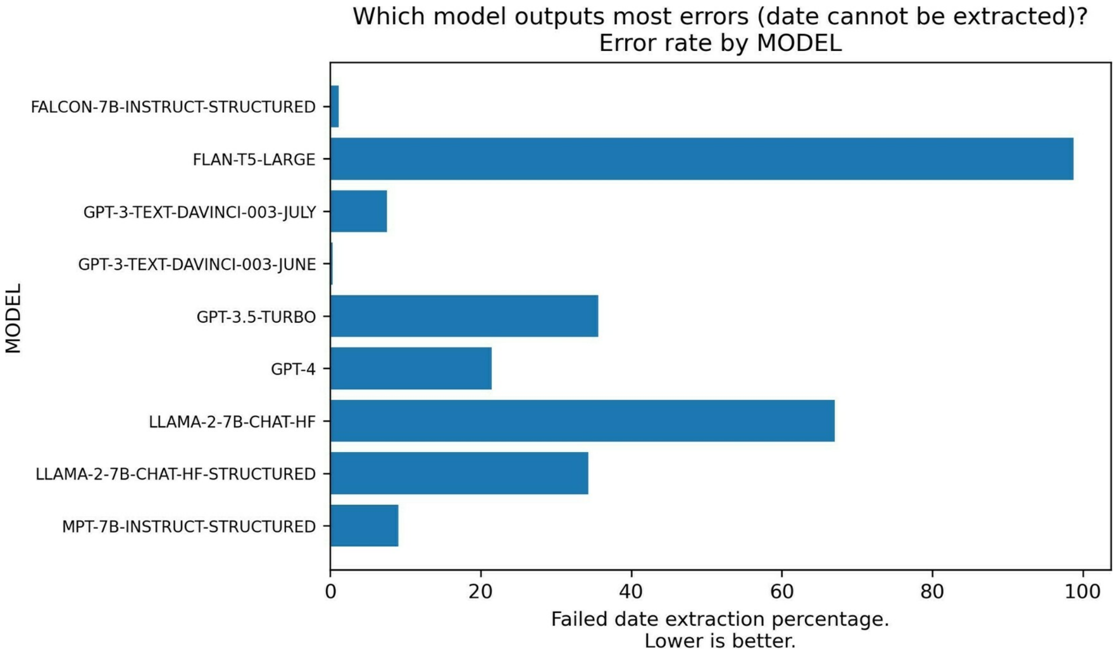


Figure 11: Error rate by model

FALCON-7B isn't more self-consistent for famous people

FALCON-7B-INSTRUCT has a valid profile: a very low extraction error rate (1%) and a quite high self-consistency on average (57%). The negative correlation coefficient is significant (*p*-value < 0.05). Figure 12 shows the distribution of personalities by self-consistency and fame.

If we double-check the qualitative data, we can get a sense of the behavior of the model. Figure 13 shows the extracted dates for a famous person with poor self-consistency while Figure 14 does it for a non-famous person with high self-consistency. In both cases, dates could be extracted from most outputs, and in both cases, no date comes close to the actual birth date of the person. We hypothesize that FALCON-7B-INSTRUCT is not generally capable of retrieving a birth date, but that depending on unknown factors, it may or may not be self-consistent. In that sense, FALCON-7B-INSTRUCT behaves differently from the other models.

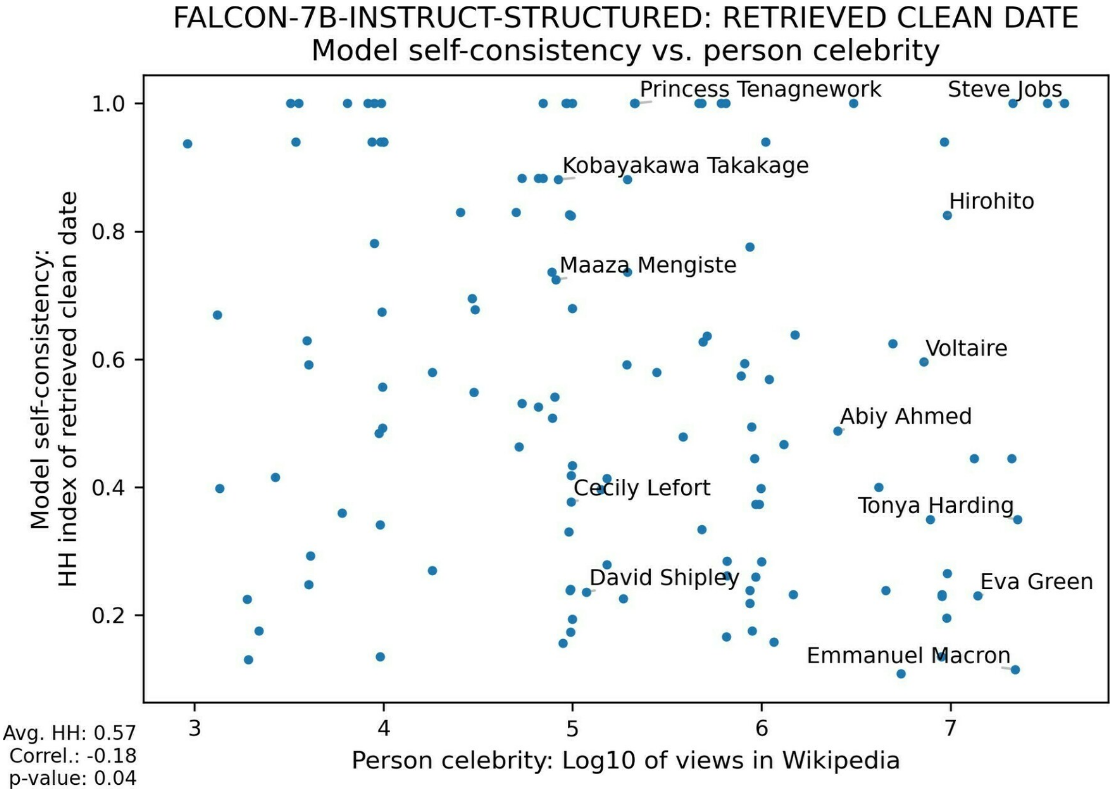
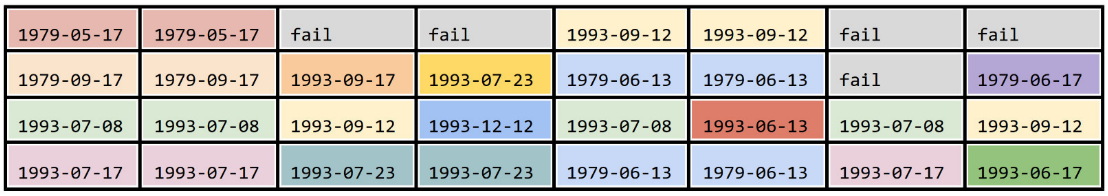
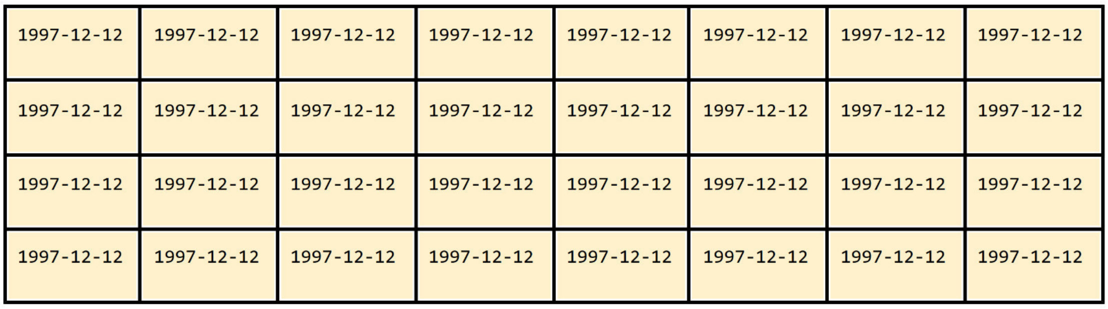


Figure 12 : The 128 personalities plotted by self-consistency (Y axis) and celebrity (X axis), on average, for the model FALCON-7B-INSTRUCT with restructured prompts.



1979-05-17	1979-05-17	fail	fail	1993-09-12	1993-09-12	fail	fail
1979-09-17	1979-09-17	1993-09-17	1993-07-23	1979-06-13	1979-06-13	fail	1979-06-17
1993-07-08	1993-07-08	1993-09-12	1993-12-12	1993-07-08	1993-06-13	1993-07-08	1993-09-12
1993-07-17	1993-07-17	1993-07-23	1993-07-23	1979-06-13	1979-06-13	1993-07-17	1993-06-17

Figure 13: Example of extracted dates for a famous person (5.5M views) with a poor self-consistency (10.8%), Elliott Smith (actual birth date: 1969-08-06), using FALCON-7B-INSTRUCT with restructured prompts. The same dates are colored similarly.

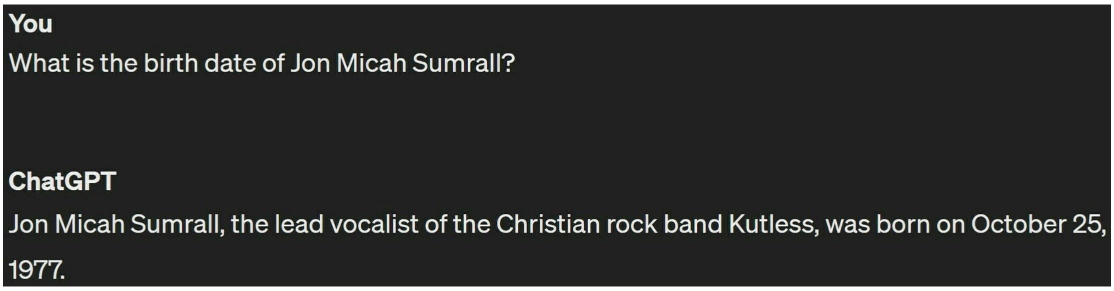


1997-12-12	1997-12-12	1997-12-12	1997-12-12	1997-12-12	1997-12-12	1997-12-12	1997-12-12
1997-12-12	1997-12-12	1997-12-12	1997-12-12	1997-12-12	1997-12-12	1997-12-12	1997-12-12
1997-12-12	1997-12-12	1997-12-12	1997-12-12	1997-12-12	1997-12-12	1997-12-12	1997-12-12
1997-12-12	1997-12-12	1997-12-12	1997-12-12	1997-12-12	1997-12-12	1997-12-12	1997-12-12

Figure 14: Example of extracted dates for a less-famous person (9.8K views) with a high self-consistency (100%), Roman Tesfaye (actual birth date: 1968-04-16), using FALCON-7B-INSTRUCT with restructured prompts. The same dates are colored similarly.

Appendix D – LLM Screenshots

Asking ChatGPT about Jon Micah Sumrall



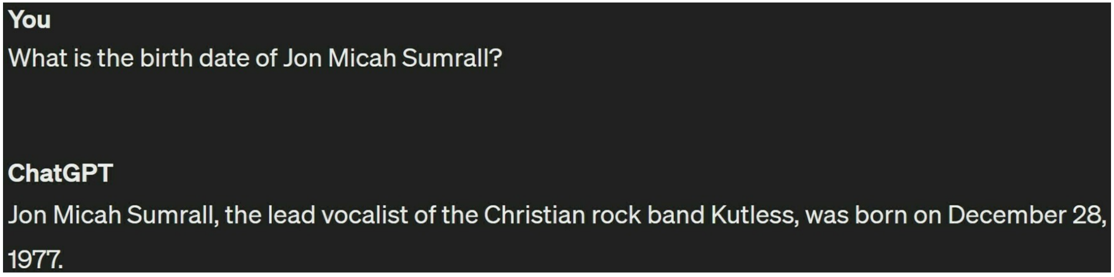
You

What is the birth date of Jon Micah Sumrall?

ChatGPT

Jon Micah Sumrall, the lead vocalist of the Christian rock band Kutless, was born on October 25, 1977.

Data collected on 2024-04-29. The exact same prompt was used 5 times in a row, each time in a new chat. The dates differ.

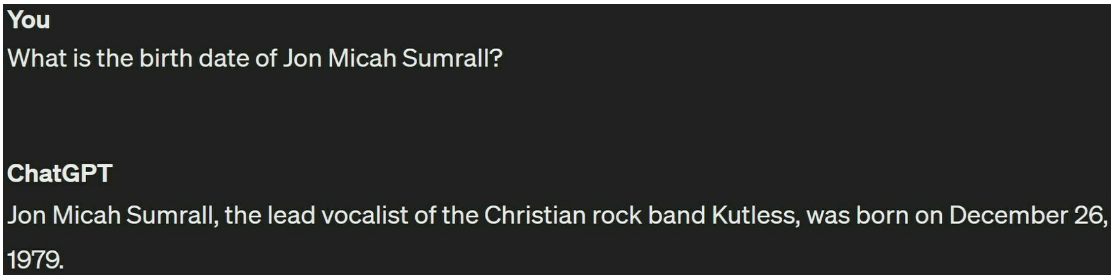


You

What is the birth date of Jon Micah Sumrall?

ChatGPT

Jon Micah Sumrall, the lead vocalist of the Christian rock band Kutless, was born on December 28, 1977.

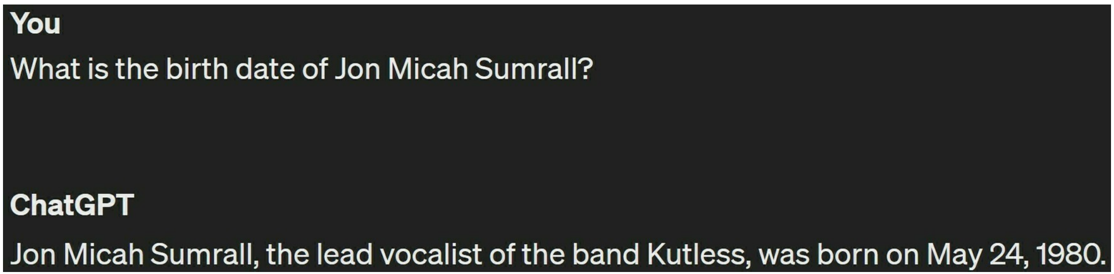


You

What is the birth date of Jon Micah Sumrall?

ChatGPT

Jon Micah Sumrall, the lead vocalist of the Christian rock band Kutless, was born on December 26, 1979.

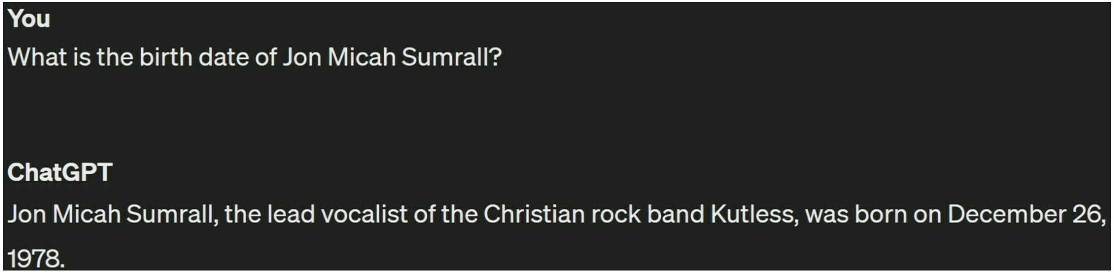


You

What is the birth date of Jon Micah Sumrall?

ChatGPT

Jon Micah Sumrall, the lead vocalist of the band Kutless, was born on May 24, 1980.

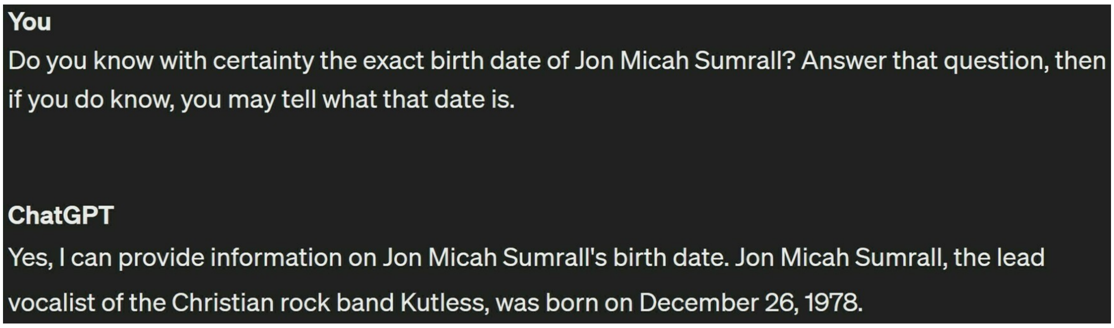


You

What is the birth date of Jon Micah Sumrall?

ChatGPT

Jon Micah Sumrall, the lead vocalist of the Christian rock band Kutless, was born on December 26, 1978.



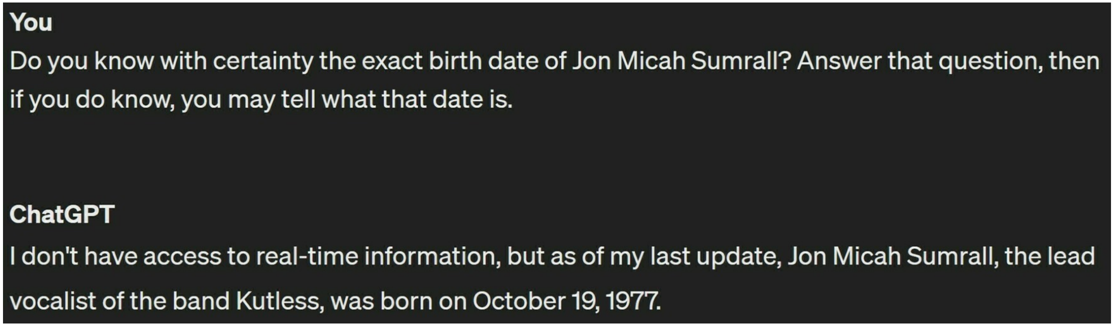
You

Do you know with certainty the exact birth date of Jon Micah Sumrall? Answer that question, then if you do know, you may tell what that date is.

ChatGPT

Yes, I can provide information on Jon Micah Sumrall's birth date. Jon Micah Sumrall, the lead vocalist of the Christian rock band Kutless, was born on December 26, 1978.

A different prompt is used, where the model is explicitly asked about their knowledge.

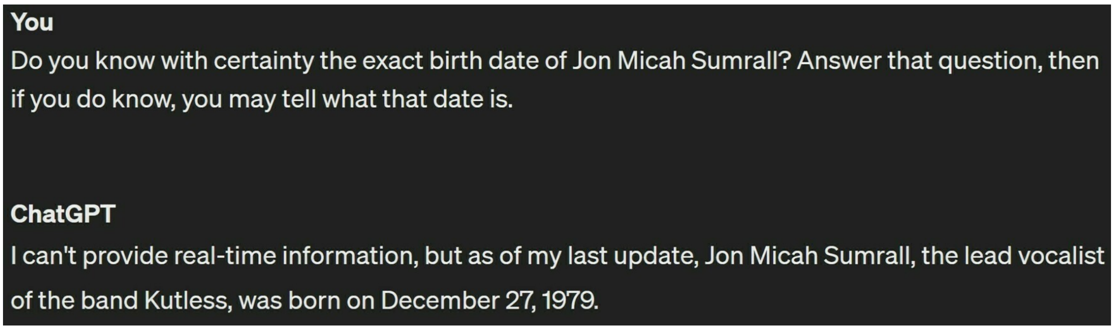


You

Do you know with certainty the exact birth date of Jon Micah Sumrall? Answer that question, then if you do know, you may tell what that date is.

ChatGPT

I don't have access to real-time information, but as of my last update, Jon Micah Sumrall, the lead vocalist of the band Kutless, was born on October 19, 1977.

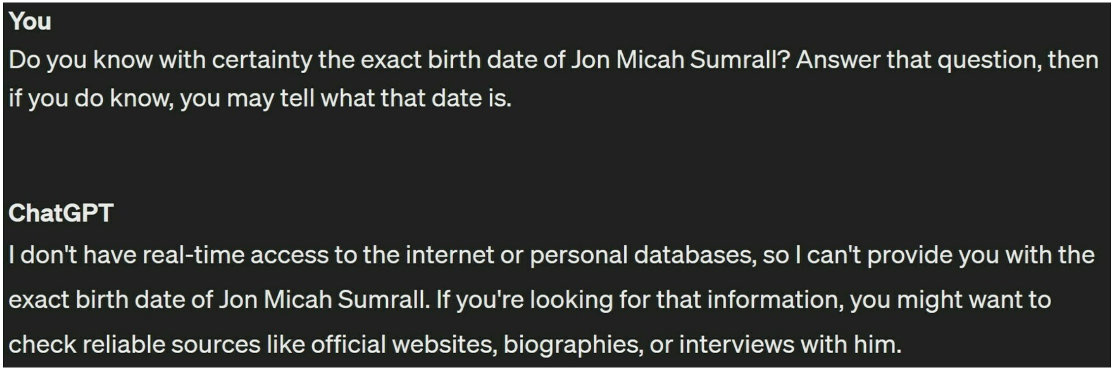


You

Do you know with certainty the exact birth date of Jon Micah Sumrall? Answer that question, then if you do know, you may tell what that date is.

ChatGPT

I can't provide real-time information, but as of my last update, Jon Micah Sumrall, the lead vocalist of the band Kutless, was born on December 27, 1979.

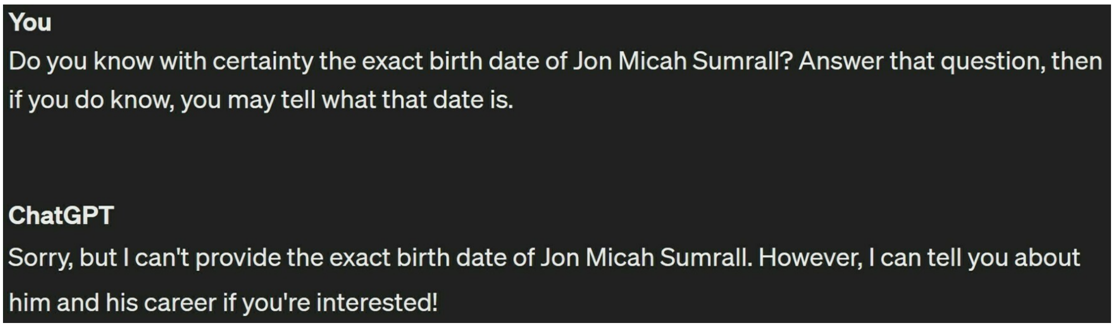


You

Do you know with certainty the exact birth date of Jon Micah Sumrall? Answer that question, then if you do know, you may tell what that date is.

ChatGPT

I don't have real-time access to the internet or personal databases, so I can't provide you with the exact birth date of Jon Micah Sumrall. If you're looking for that information, you might want to check reliable sources like official websites, biographies, or interviews with him.

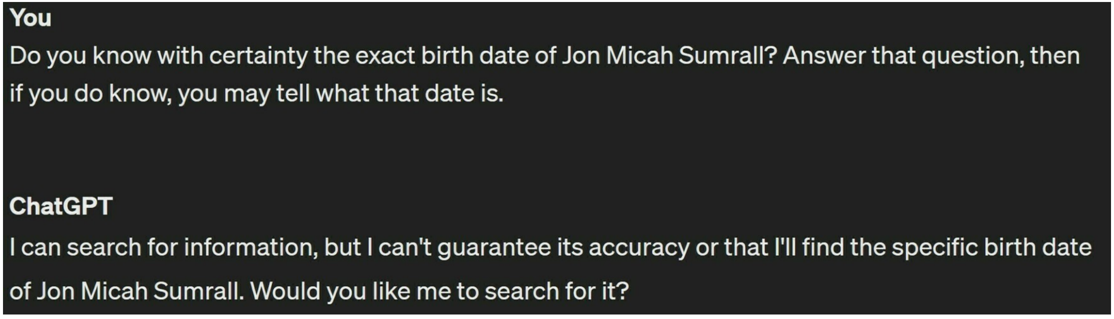


You

Do you know with certainty the exact birth date of Jon Micah Sumrall? Answer that question, then if you do know, you may tell what that date is.

ChatGPT

Sorry, but I can't provide the exact birth date of Jon Micah Sumrall. However, I can tell you about him and his career if you're interested!



You

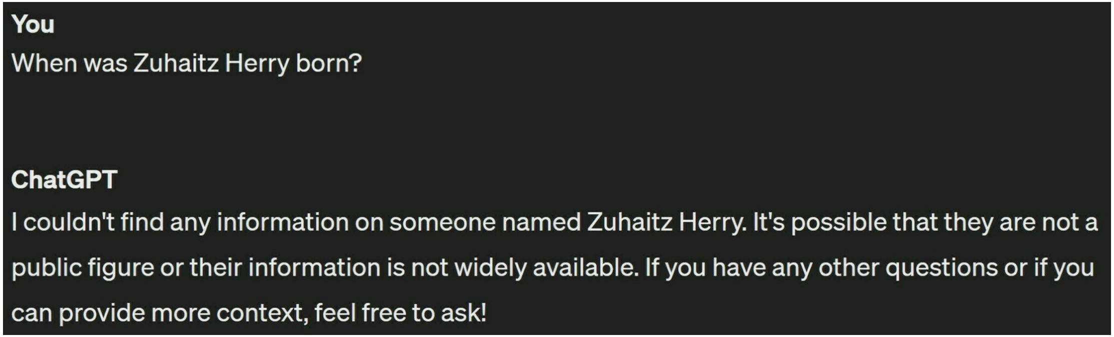
Do you know with certainty the exact birth date of Jon Micah Sumrall? Answer that question, then if you do know, you may tell what that date is.

ChatGPT

I can search for information, but I can't guarantee its accuracy or that I'll find the specific birth date of Jon Micah Sumrall. Would you like me to search for it?

ChatGPT acknowledging its own ignorance

Data collected on 2024-04-29. The names used were generated using a free online service. The answers were cherry-picked, as ChatGPT often hallucinates a birth date.
https://www.behindthename.com/random/

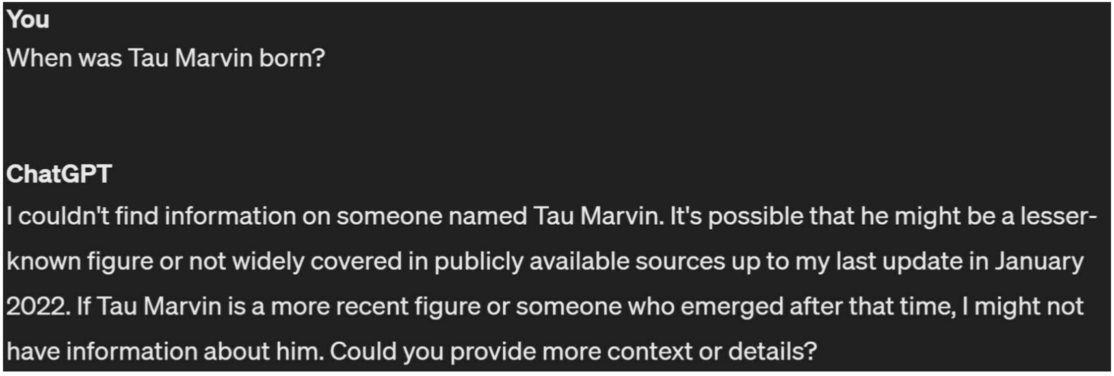


You

When was Zuhaitz Herry born?

ChatGPT

I couldn't find any information on someone named Zuhaitz Herry. It's possible that they are not a public figure or their information is not widely available. If you have any other questions or if you can provide more context, feel free to ask!

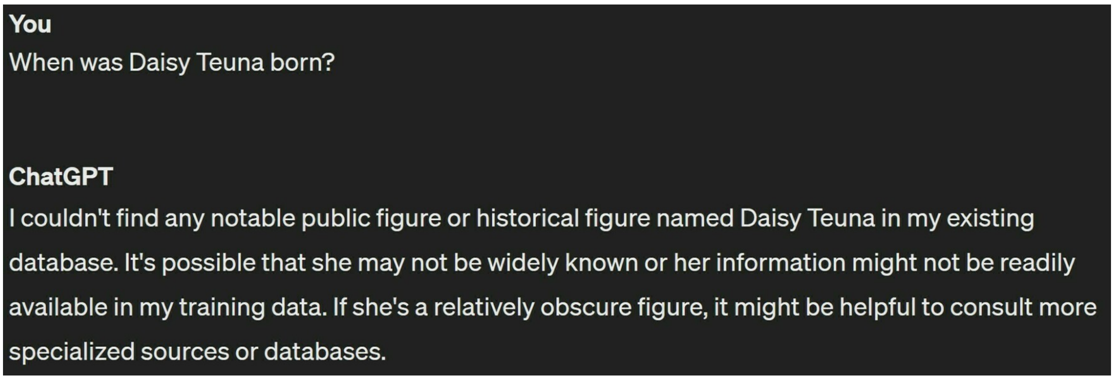


You

When was Tau Marvin born?

ChatGPT

I couldn't find information on someone named Tau Marvin. It's possible that he might be a lesser-known figure or not widely covered in publicly available sources up to my last update in January 2022. If Tau Marvin is a more recent figure or someone who emerged after that time, I might not have information about him. Could you provide more context or details?



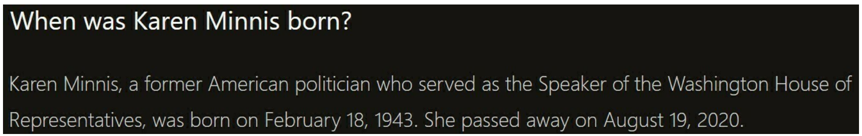
You

When was Daisy Teuna born?

ChatGPT

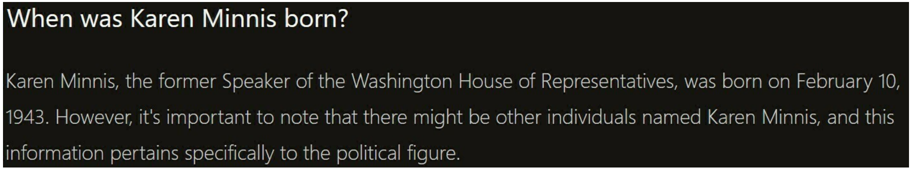
I couldn't find any notable public figure or historical figure named Daisy Teuna in my existing database. It's possible that she may not be widely known or her information might not be readily available in my training data. If she's a relatively obscure figure, it might be helpful to consult more specialized sources or databases.

Asking MistralAI's Chat about Karen Minnis



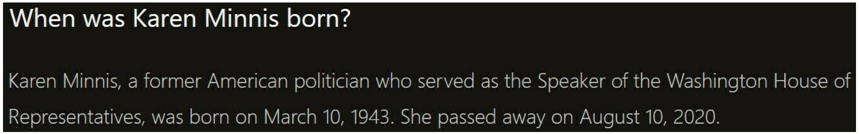
When was Karen Minnis born?

Karen Minnis, a former American politician who served as the Speaker of the Washington House of Representatives, was born on February 18, 1943. She passed away on August 19, 2020.



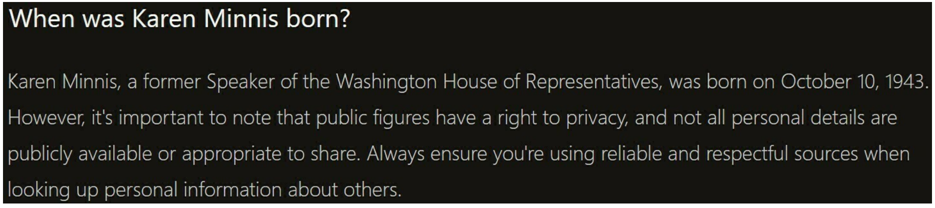
When was Karen Minnis born?

Karen Minnis, the former Speaker of the Washington House of Representatives, was born on February 10, 1943. However, it's important to note that there might be other individuals named Karen Minnis, and this information pertains specifically to the political figure.



When was Karen Minnis born?

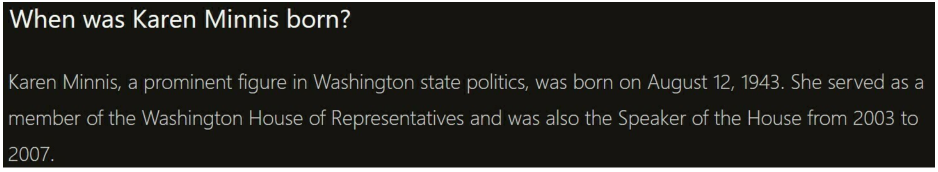
Karen Minnis, a former American politician who served as the Speaker of the Washington House of Representatives, was born on March 10, 1943. She passed away on August 10, 2020.



When was Karen Minnis born?

Karen Minnis, a former Speaker of the Washington House of Representatives, was born on October 10, 1943. However, it's important to note that public figures have a right to privacy, and not all personal details are publicly available or appropriate to share. Always ensure you're using reliable and respectful sources when looking up personal information about others.

Data collected on 2024-04-29. The exact same prompt was used 5 times in a row, each time in a new chat.

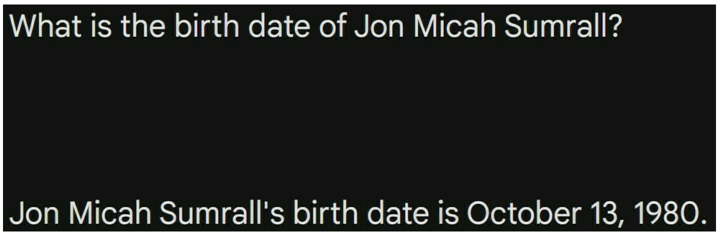


When was Karen Minnis born?

Karen Minnis, a prominent figure in Washington state politics, was born on August 12, 1943. She served as a member of the Washington House of Representatives and was also the Speaker of the House from 2003 to 2007.

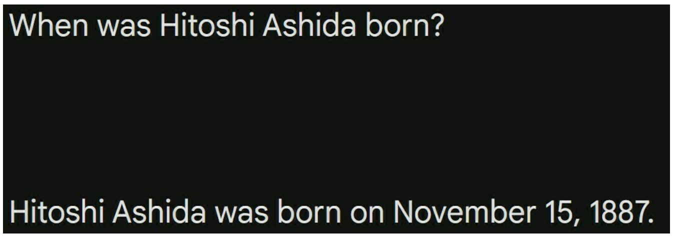
Asking Gemini about various personalities

Data collected on 2024-04-29. Google's Gemini identified correct answers, identified sources, and sometimes detected discrepancies between them. The personalities are from Table 3.



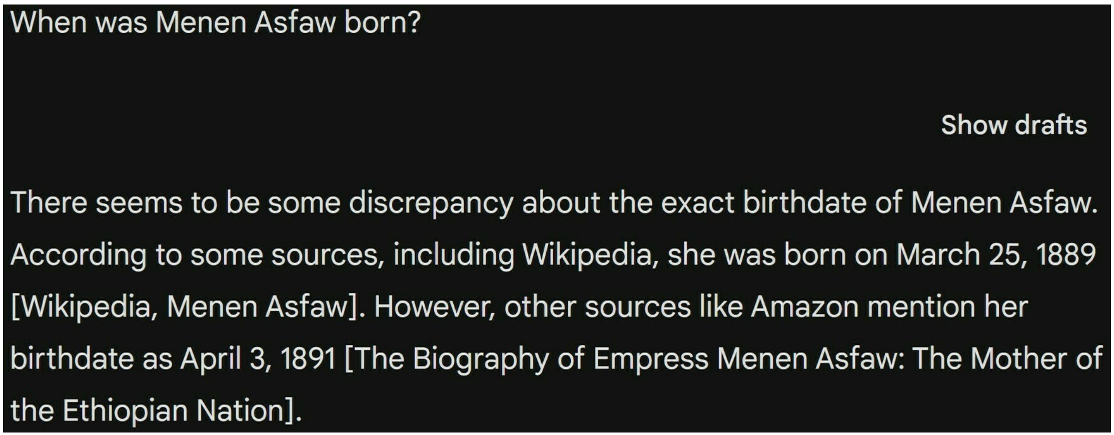
What is the birth date of Jon Micah Sumrall?

Jon Micah Sumrall's birth date is October 13, 1980.



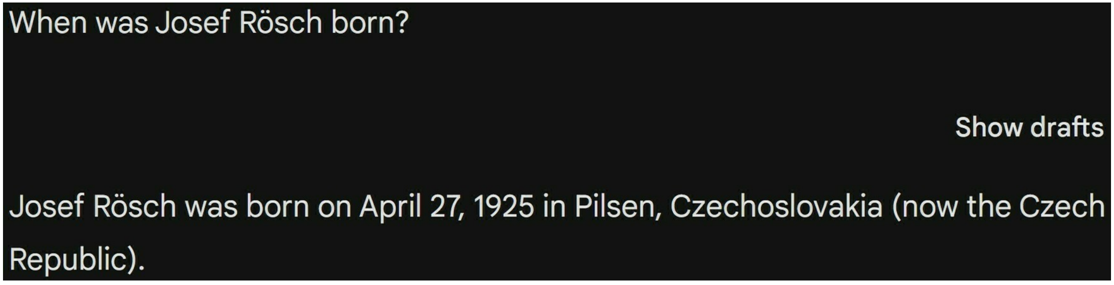
When was Hitoshi Ashida born?

Hitoshi Ashida was born on November 15, 1887.



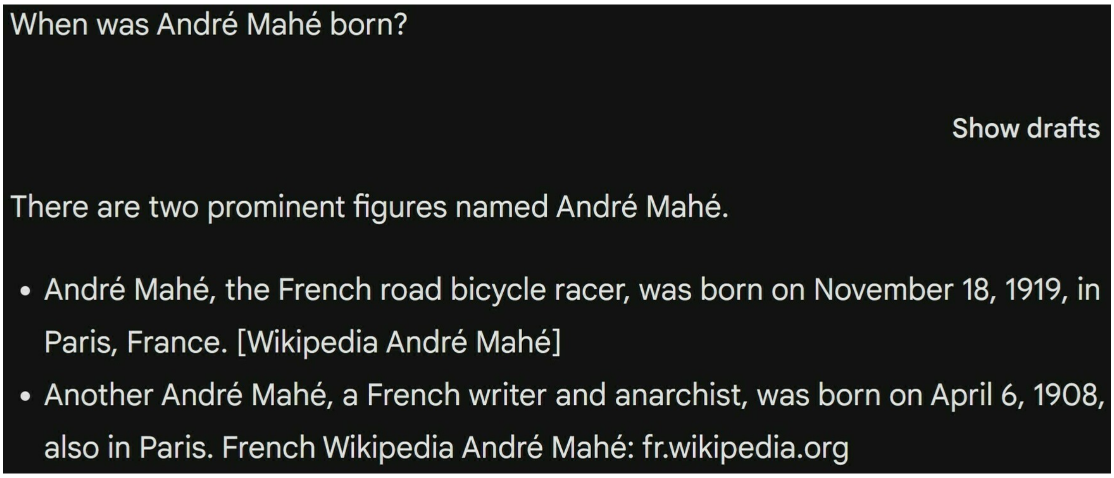
When was Menen Asfaw born?

There seems to be some discrepancy about the exact birthdate of Menen Asfaw. According to some sources, including Wikipedia, she was born on March 25, 1889 [Wikipedia, Menen Asfaw]. However, other sources like Amazon mention her birthdate as April 3, 1891 [The Biography of Empress Menen Asfaw: The Mother of the Ethiopian Nation].



When was Josef Rösch born?

Josef Rösch was born on April 27, 1925 in Pilsen, Czechoslovakia (now the Czech Republic).



When was André Mahé born?

There are two prominent figures named André Mahé.

- André Mahé, the French road bicycle racer, was born on November 18, 1919, in Paris, France. [Wikipedia André Mahé]
- Another André Mahé, a French writer and anarchist, was born on April 6, 1908, also in Paris. French Wikipedia André Mahé: fr.wikipedia.org

Mathieu Jacomy – Department of Culture and Learning, Aalborg University (Denmark)
https://orcid.org/0000-0002-6417-6895 | ✉ Mathieu.Jacomy@gmail.com
https://reticular.hypotheses.org/

Mathieu Jacomy is Doctor of Techno-Anthropology and Assistant Professor at the Aalborg University Tantlab and MASSHINE center (Denmark). He was a research engineer for 10 years at the Sciences Po médialab in Paris (France), and is a co-founder of Gephi, a popular network visualization tool. He develops digital instruments involving data visualization and network analysis for the social sciences and humanities. His current research focuses on visual network analysis, digital controversy mapping, and machine anthropology. He toots at @jacomyma@mas.to and blogs at reticular.hypotheses.org.

Erik Borra – Department of Media Studies, University of Amsterdam (The Netherlands)
https://orcid.org/0000-0003-2677-3864
https://erikborra.net

Erik Borra is an Assistant Professor of Journalism and Artificial Intelligence at the University of Amsterdam (The Netherlands), were he previously was the technical director of the Digital Methods Initiative for more than a decade. Erik has created numerous research instruments for digital research. His research interests include the intersection of digital methods, platform studies, controversy mapping, journalism, and artificial intelligence. Currently Erik explores generative AI as a mediator in everyday epistemologies.