The Evaluation of Artificial Intelligence as a Prediction Problem

September 2024

Author: Wout Schellaert

Advisors: José Hernández-Orallo
Fernando Martínez-Plumed

Summary

Artificial intelligence (AI) is becoming more and more relevant to our daily lives. Countless products and services rely on some sort of machine cognition to function. Given this increased relevance, the evaluation of AI systems, i.e. measuring how well they perform or how intelligent they are, is however not in the state we would like it to be. Both in practice and in theory, evaluation is lacking in various ways. Performance estimations suffer in accuracy under distribution shift and test set contamination. Given a new instance of a task, most evaluation procedures do not provide a granular performance estimate tailored to that instance. And with the advent of general-purpose AI (GPAI), the scope of evaluation has increased significantly, associating a heavy data and logistical burden with common evaluation methodology, and entailing an unknown distribution of tasks to measure performance for.

Motivated by the importance of prediction in the philosophy of science, the relevance of prediction to the practices of evaluation in other fields such as animal cognition and psychometrics, and the role of prediction as a central language in the field of machine learning itself, we frame the evaluation of artificial intelligence as a prediction problem. We develop a formal framework conceptualising evaluation procedures as learning algorithms that produce performance-predicting models from empirical data. We organise various existing evaluation techniques and applications using this framework, identifying connections between previously unrelated ideas and highlighting gaps in current approaches.

This framework then helped us reframe the challenges of evaluation in the language of statistics and machine learning, providing a solution to the refinement

problem by conditioning evaluation results on the input variable, which also partially addresses the distribution shift problem. For challenges that remain unsolved. e.g. out-of-distribution prediction and the large scope of evaluating GPAI systems, clear analogies to machine learning literature can be drawn, providing a way forward through scaling up data, focusing on generalisation, or introducing meta-learning for evaluation.

Additionally, two empirical studies are presented. The first is an investigation of "assessor models", a newly developed machine learning technique for granularly predicting AI system performance, where the experimental setting is focused on large language models (LLMs) and the factors influencing the predictability of their performance. We find that refined instance-level score estimation is possible, out-of-distribution score estimation is consistently hard, and that the use of multi-task and multi-system data can improve evaluation accuracy

The second study again investigates language model performance, but now from the perspective of human users doing score prediction in their direct interaction with AI systems. Starting from human-derived notions of difficulty, we analyse performance, question avoidance, prompt sensitivity, and human supervision across five different tasks over varying difficulty levels. We also add two other dimensions: that of scaling language models, i.e. making them bigger in terms of parameters and ingested data, and that of shaping them, making them more instructable and easier to use. Over more than thirty models from three different language model families (GPT, Llama, and BLOOM), we find that while human difficulty correlates well with LLM performance, there is too little of a step shape in the difficulty-performance plot to facilitate confident prediction. There is no region –regardless how low the difficulty– where performance is perfect, as newer models improve by gaining ground on the medium and hard instances, rather than the easy ones.

We conclude that a predictive interpretation of evaluation brings together different techniques and applications of evaluation. By drawing from the rich literature of machine learning and statistics, we can solve several fundamental issues in evaluation theory, while making progress in several others, allowing the science and practice of evaluation to more naturally co-evolve with the advances of artificial intelligence.

Resumen

La inteligencia artificial (IA) es cada vez más importante en nuestra vida cotidiana. Innumerables productos y servicios dependen de algún tipo de cognición artificial para funcionar. Dada su creciente relevancia, la evaluación de los sistemas de IA, es decir, medir lo bien que funcionan o lo inteligentes que son, no se encuentra, sin embargo, en el estado que nos gustaría. Tanto en la práctica como en la teoría, la evaluación adolece de varias carencias. La precisión de las estimaciones de rendimiento se ve afectada por cambios en la distribución y la contaminación del conjunto de pruebas. Dada una nueva instancia de una tarea, la mayoría de los procedimientos de evaluación no proporcionan una estimación granular del rendimiento adaptada a esa instancia. Además, con la llegada de la IA de propósito general (GPAI), el alcance de la evaluación ha aumentado significativamente, lo que conlleva una pesada carga logística y de datos para las metodologías de evaluación comunmente aplicadas así como una distribución desconocida de tareas sobre las que medir el rendimiento.

Motivados por la importancia de la predicción en la filosofía de la ciencia, por la relevancia de la predicción para las prácticas de evaluación en otros campos, como la cognición animal y la psicometría, y por el papel de la predicción como lenguaje central en el propio campo del aprendizaje automático, enmarcamos la evaluación de la inteligencia artificial como un problema de predicción. Desarrollamos un marco formal que conceptualiza los procedimientos de evaluación como algoritmos de aprendizaje que producen modelos de predicción del rendimiento a partir de datos empíricos. Organizamos diversas técnicas y aplicaciones de evaluación existentes utilizando este marco, en el que se identifican conexiones entre

ideas hasta entonces inconexas y se ponen de relieve las lagunas de los enfoques actuales.

Este marco nos ha ayudado a replantearnos los retos de la evaluación en el lenguaje de la estadística y el aprendizaje automático, aportando una solución al problema del refinamiento al condicionar los resultados de la evaluación a la variable de entrada, lo que también aborda parcialmente los problemas de cambio de distribución. Para los retos que siguen sin resolverse, como la predicción fuera de distribución y el amplio alcance de la evaluación de los sistemas GPAI, pueden trazarse claras analogías con la literatura sobre aprendizaje automático. Esto proporciona un camino a seguir mediante el escalado de los datos, centrándose en la generalización, o introduciendo el meta-aprendizaje para la evaluación.

Además, se presentan dos estudios empíricos. El primero es una investigación de los «assessor models», una técnica de aprendizaje automático desarrollada recientemente para predecir de forma granular el rendimiento de los sistemas de IA, en la que el escenario experimental se centra en los grandes modelos de lenguaje (LLM, por sus siglas en ingés) y en los factores que influyen en la predictibilidad de su rendimiento. Se constata que es posible realizar una estimación refinada de la evaluación a nivel de instancia, que la estimación de la evaluación fuera de la distribución es sistemáticamente difícil y que el uso de datos multitarea y multisistema puede mejorar la precisión de la evaluación.

El segundo estudio investiga de nuevo el rendimiento de los modelos de lenguaje, pero en esta ocasión desde la perspectiva de los usuarios humanos que realizan predicciones de evaluación en su interacción directa con los sistemas de IA. Partiendo de nociones de dificultad derivadas de los humanos, se analiza el rendimiento, la evitación de preguntas, la sensibilidad a las instrucciones proporcionadas y la supervisión humana en cinco tareas diferentes con distintos niveles de dificultad. También añadimos otras dos dimensiones: el escalado de los modelos de lenguaje, es decir, hacerlos más grandes en términos de parámetros y datos ingeridos, y su conformación, haciéndolos más instruibles y fáciles de usar. En más de treinta modelos de tres familias diferentes de modelos de lenguaje (GPT, Llama y BLOOM), observamos que, si bien la dificultad humana se correlaciona bien con el rendimiento de los LLM, la forma escalonada del gráfico dificultad-rendimiento es demasiado pequeña para facilitar una predicción fiable. No existe una región –ni siquiera reduciendo la dificultad– en la que el rendimiento sea perfecto, y los modelos más nuevos mejoran ganando terreno en las instancias medias y difíciles, más que en las fáciles.

Llegamos a la conclusión de que una interpretación predictiva de la evaluación permite unificar una variedad de técnicas y aplicaciones de evaluación. Basándose

en la extensa literatura del aprendizaje automático y la estadística, es posible resolver además varias cuestiones fundamentales de la teoría de la evaluación, al tiempo que avanza en otras, permitiendo que la ciencia y la práctica de la evaluación coevolucionen de forma más natural con los avances de la inteligencia artificial.

Resum

La intel·ligència artificial (IA) és cada vegada més important en la nostra vida quotidiana. Innombrables productes i serveis depenen d'alguna mena de cognició artificial per a funcionar. Donada aquesta creixent rellevància, l'avaluació dels sistemes d'IA, és a dir, mesurar com de bé funcionen o com d'intel·ligents són, no es troba, però, en l'estat que ens agradaria. Tant en la pràctica com en la teoria, l'avaluació pateix de diverses mancances. La precisió de les estimacions de rendiment es veu afectada per canvis en la distribució i la contaminació del conjunt de proves. Donada una nova instància d'una tasca, la majoria dels procediments d'avaluació no proporcionen una estimació granular del rendiment adaptada a eixa instància. A més, amb l'arribada de la IA de propòsit general (GPAI), l'abast de l'avaluació ha augmentat significativament, la qual cosa comporta una pesada càrrega logística i de dades per a les metodologies d'avaluació comunament aplicades així com una distribució desconeguda de tasques sobre les quals mesurar el rendiment.

Motivats per la importància de la predicció en la filosofia de la ciència, per la rellevància de la predicció per a les pràctiques d'avaluació en altres camps, com la cognició animal i la psicometria, i pel paper de la predicció com a llenguatge central en el propi camp de l'aprenentatge automàtic, emmarquem l'avaluació de la intel·ligència artificial com un problema de predicció. Desenvolupem un marc formal que conceptualitza els procediments d'avaluació com a algorismes d'aprenentatge que produeixen models de predicció del rendiment a partir de dades empíriques. Organitzem diverses tècniques i aplicacions d'avaluació existents utilitzant aquest marc, en el qual s'identifiquen connexions entre idees fins llavors inconnexes i es posen en relleu les llacunes dels enfocaments actuals.

Aquest marc ens ha ajudat a replantejar-nos els reptes de l'avaluació en el llenguatge de l'estadística i l'aprenentatge automàtic, aportant una solució al problema del refinament en condicionar els resultats de l'avaluació a la variable d'entrada, la qual cosa també aborda parcialment els problemes de canvi de distribució. Per als reptes que segueixen sense resoldre's, com la predicció fora de distribució i l'ampli abast de l'avaluació dels sistemes GPAI, poden traçar-se clares analogies amb la literatura sobre aprenentatge automàtic. Això proporciona un camí a seguir mitjançant l'escalat de les dades, centrant-se en la generalització, o introduint el meta-aprenentatge per a l'avaluació.

A més, es presenten dos estudis empírics. El primer és una investigació dels «assessors models», una tècnica d'aprenentatge automàtic desenvolupada recentment per a predir de manera granular el rendiment dels sistemes d'IA, en la qual l'escenari experimental se centra en els grans models de llenguatge (LLM, per les seues sigles en anglés) i en els factors que influeixen en la predictibilitat del seu rendiment. Es constata que és possible realitzar una estimació refinada de l'avaluació a nivell d'instància, que l'estimació de l'avaluació fora de la distribució és sistemàticament difícil i que l'ús de dades multitasca i multisistema pot millorar la precisió de l'avaluació.

El segon estudi investiga de nou el rendiment dels models de llenguatge, però en aquesta ocasió des de la perspectiva dels usuaris humans que realitzen prediccions d'avaluació en la seua interacció directa amb els sistemes de IA. Partint de nocions de dificultat derivades dels humans, s'analitza el rendiment, l'evitació de preguntes, la sensibilitat a les instruccions proporcionades i la supervisió humana en cinc tasques diferents amb diferents nivells de dificultat. També afegim altres dos dimensions: l'escalat dels models de llenguatge, és a dir, fer-los més grans en termes de paràmetres i dades ingerides, i la seua conformació, fent-los més instruïbles i fàcils d'usar. En més de trenta models de tres famílies diferents (GPT, Llama i BLOOM), observem que, si bé la dificultat humana es correlaciona bé amb el rendiment dels LLM, la forma escalonada del gràfic dificultat-rendiment és massa suau per a facilitar una predicció fiable. No existeix una regió –ni tan sols reduïnt la dificultat– en la qual el rendiment siga perfecte, i els models més nous milloren guanyant terreny en les instàncies mitjanes i difícils, més que en les fàcils.

Arribem a la conclusió que una interpretació predictiva de l'avaluació permet unificar una varietat de tècniques i aplicacions d'avaluació. Basant-se en l'extensa literatura de l'aprenentatge automàtic i l'estadística, és possible resoldre a més diverses qüestions fonamentals de la teoria de l'avaluació, al mateix temps que avança en unes altres, permetent que la ciència i la pràctica de l'avaluació coevolucionen de forma més natural amb els avanços de la intel·ligència artificial.

Acknowledgements

Advisors

To José, Nando... My first kind words are for you, as it is with you that my wonderful doctoral journey started. I often remember –with extra joy now– our first interaction, how it was all so simple. After a long-awaited but sudden realisation that I really did want to pursue research, I went ahead and constructed a haphazard list of scientists whose work fascinated me, finding your name on top. And then, I simply sent you an e-mail. Some months later, I'm moving away from friends and family to a city unknown, still green behind my ears. You read my rambling thoughts with an open mind, answered quickly, and in one fell swoop convinced me with your kindness and sincerity. I never felt the need to mail the runner-ups.

You were flexible, timely, and supportive. You were bold, trusting, and fun. I can still recollect vividly those first weeks in Valencia, in between Fallas and a frantic search for a place to live, and that unexpected call to write a paper from scratch, with a deadline two weeks hence. What a start... Apart from the thrill and the surprised wonder whether all scientific processes was like this, I recall how from day one my opinions and thoughts were respected, how you invested your time, how, even in confrontation, the person –not the product–, came first. In many ways, things slowed down from there, in any many others, they only sped up, together aiming for the stars. In the process, you did not act like masters or bosses, but like companions on the road, whose advise and help made the journey more worthwhile than the destination. Thank you for giving me the freedom to make my own mistakes. Thank your being amicable the whole way through.

To Nando, you deserve more credit for picking up the pieces of work that fall in the cracks. You put others first. Behind that tough guy vibe is a mellow heart of gold.

To José, your knowledge, dedication, and wisdom are immensely impressive. You dance in the scientific world but stay with your feet on the ground. You are an inspiration.

You made a fantastic pair, and I could not have ended up anywhere better.

Financial and Logistical Support

I want to thank the UPV, the Doctoral School, the Vicerectorate of Investigation, the European Social Fund, and the Spanish Ministry of Education, Culture, Universities and Employment for funding my doctoral research through grant agreements DOCEMPR21, DOCEMPR22 and DOCEMPR23, as well as SOCERFAB SALUD S.R.L. for being an industrial partner. An additional thanks goes to Santiago Escobar, who funded the cracks in my coverage.

I also acknowledge support from EU's Horizon 2020 Research and Innovation Programme, which made a long and wonderful research stay possible under grant agreement No. 952215 (TAILOR).

Thank you also to Lama Ahmad, Miles Brundage, and OpenAI for generously providing funded research access to the future. Thank you to European Commission's JRC and Emilia Gomez for valuing my expertise, and financially supporting my research. It made a world of difference. Thank you to Stuart Elliot, Abel Baret, and the OECD, for multiple years of research partnership.

I also need to convey my gratitude to the people running the HPC at VRAIN for making me feel like one of the cool kids and having compute to play with, and to those who thanklessly maintain open-source software. In no particular order, and by no means exhaustive, but a heartfelt thank you to the people behind Latex, Nix, Huggingface, and Matplotlib.

And Jan, thank you for your pool, on that one very hot summer day.

Reviewers

Thank you to the reviewers, Ricardo Bastos Cavalcante Prudencio and Maria Riveiro Carballa, for taking the time, for your feedback, and for your kind words. I glowed up.

Colleagues and Collaborators

Thank you to Ronan Hamon and Ignacio Sanchez for hosting me in delightful Ispra, having a flexible schedule, and giving me freedom to pursue the research I wanted.

Thank you, Yvan Saeys. First, for giving me complete freedom during my master's thesis, which catalysed my passion for the scientific endeavour, and second, for hosting me at the VIB for a year-long research stay. It made a lot of difference in a lot of different ways. Thank you too to all Dambini's, for accepting me so quickly in your circles, for your kindness and your companionship; you added a lot of colour to my 2024.

Thank you to the LCFI and Cambridge crowd, Kozzy Voudouris, John Burden, Lorenzo Pacchiardi, Ryan Burnell, Danaja Rutar, Marko Tešić, Benjamin Slater, Lucy Cheke, for the years of enthusiastic collaboration from across the channel.

Thank you to those who do not fit a box, Niall Donnelly, Peter Flach, Manuel Cebrian, Tomer D. Ullman, Tony Cohn, Pablo Antonio Moreno Casares, but nonetheless have enriched my research adventure with advice, collaboration, or collegiality.

Thank you to the wide DMIP research group and all it's visitors, for the companionship and help over the years. Thank you Carlos Monserrat-Aranda, María José Ramírez, Cristina Padró, Nikolai Sanamrad, Raul Fabra, Joe Castellano, David Castellano Falcón, Diogo Freitas, Daniel Romero Alvarado, Jan Arne Telle, Brigt Håvardstun, Dario Garigliotti.

Thank you, team Pinocchio, for the adventure that delivered the crown to this dissertation. Lex, Yael, you will achieve great things. If you would want to of course; don't let pressure take the fun out of anything.

Thank you Cèsar, for taking us on a wonderful holiday in the Pyrenees, for organising so many hiking trips, for explaining fronton, for being your charismatic self, always open and inviting.

Thank you Yue, for persisting in having conversations, even when the language made it difficult. You were great fun, always jovial and kind. Sharing in your wisdom and culture enriched me in many different ways. Your resilience inspires. May you have good fortune in the rest of your live, and may our paths cross again.

Behzad, I don't know where to start. Our lab was never big, but you fill so many shoes it did not need to be. I wish I could reconcile our two worlds. I wish I could convince you to be less humble. You are so immensely brave, caring, smart, funny. Every conversation was delight, sweet and complete, the Persian way. Mahnaz, Elena and you are like family, I and look back fondly to our moments together.

To all, in your Valencia will forever lay a piece of my heart.

Friends and Family

Thank you to the many friends who cheered along the way, the big bears from the volleyball, the guys from high school that I see too little, the Keuns, the Zeusjes for getting me through my degree (and much other things), the DSA crew, Nando and Ana, the Broodje Ballers. Thank you Ilion, Kenneth, Noom, Elisabeth, Robbe, Wim and Margot. I do not say it enough, but I love you all very much.

Thank you to my family, you unique bunch of caring souls. Your unconditional love and support have allowed me to take on this life running and dancing, not afraid of falling, as there is always a pair of arms ready for catching. You have raised the bar for all my personal relations, both in what I expect from others, and what others can expect from me.

Tomas and Celine, in the register of our shared language, only 'schedel' conveys how much you mean to me.

Thank you to everyone who came to visit us: Bauke, Nika, Sam, Maarten Moens, Tim Vandamme, Joppe, Ma'tje, Pa'tje, Eva, Jorre, Maarten Anraed, Renee, Elisabeth, Seppe, Naomi, Jasper, Filip (the winner, with four visits!), Laszlo, David, Melis, Rémi, Amber, Carine, Sebi, Maarten Sebrechts, Margot, Tomas, Celine, Ilion, Clara, Vadim, Kenneth, Caroline, Ruben, Remco, Wim, Ann, Sarah, Joren, Fredje, Arno, Nando and Ana. With every visit, I felt like a kid again, carefree, exploring the world together, basking in the Spanish sun, getting to know each other deeper and differently, like new friends, even if we had known each other for years. I would gladly move again to a foreign ground, if it meant my days would be filled with the utter joy of so many private and dedicated days together.

Thank you ma'tje en pa'tje, for giving me all that you could. For teaching me so much, for the examples you are. Whenever life requires confidence, empathy, sacrifice, whenever I need integrity or bravery, I think of you, and I know what to do.

And lastly, my dear Isa, I can not help but reflect on you taking that big step with me, when it was all much less certain. What a different life it would have been, what courage that did require. Thank you for always betting on us, for your love and support these last years, and above all else, for every piece of beauty in the world, as to you I owe the hunger for life.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Artificial intelligence is becoming more and more relevant to our daily lives. Countless products and services rely on some sort of machine cognition to function. There are robotaxis (Marshall, 2023), deepfakes (Westerlund, 2019), and automated customer support (Hardalov et al., 2018). There are virtual assistants and virtual lovers (Brooks, 2021). From a few soundbites, scammers now clone the voices of their targets' relatives (Amezaga & Hajek, 2022), but on the flip side, when hiking in the woods an untrained ear can identify any bird sound with a free app (Merlin Bird ID, 2024). Media outlets report a rising fear of workforce replacement and a page later proudly announce a new robot-driven factory. It seems society is having (fever)-dreams of a technological singularity. Without a doubt, there is an increasingly deep integration of artificial intelligence into our digital and physical environments.

Given this increased relevance, the evaluation of these AI systems, i.e. measuring how well they perform or how intelligent they are, is not in the state we would like it to be. Both in practice and in theory, evaluation is lacking in various ways. Old problems remain while new challenges are added on top, and different evaluation procedures have disparate goals, which complicate matters. We will have much to say about these issues later, but two prominent examples that will often return are the following:

(i) If we deploy AI systems in a particular setting that is in some way different from the one in which they were tested, the evaluation metrics that had been collected quickly stop being accurate and relevant. This does not have to mean that new concepts are introduced; it is often enough that the frequency of occurrence of particular features is different. For example in the context of an AI-assisted brain tumour screening scenario, a different average age of the population or a different prevalence of tumours often constitutes enough of a change to invalidate the typical evaluation result, which generally assumes a stable proportion of cases and types of patients. We will refer to this as *distribution shift*, and it is applicable to most deployed AI systems, to the point that "many companies consider it inevitable" (Huyen, 2022, ch. 8).

(ii) A typical evaluation does not allow for granular interpretation, in the sense that when we have evaluated a system, and we now have a specific new instance of the problem (e.g. an image to classify), we do not really know how likely it is that the system will perform well. Of course we know how well it will perform on average, e.g. it has a 10% chance of failing –the same as for all instances–, but this is like a weather report saying there will be two rainy days this week, with no indication as to which ones. The problem is actually worse. In our analogy we would know that it rains e.g. about 150 days a year, without any sort of refinement with respect to months or seasons –not quite ideal for planning your barbecue. We will refer to this as the *refinement problem*.

The last example already hints at it, but our evaluations do not provide us with much *predictive power* as to where and when our AI systems will perform well. We cannot anticipate for which inputs a system might fail, and if our deployment setting deviates from our testing setting, we cannot even predict how well it will perform overall. We do not build good (mental) models of our machine learning models. For the information-gathering process that evaluation is, this is rather disappointing, as being able to make predictions has always been a central tenet in the philosophy of science. At least a few problems in AI evaluation, we think, can be helped by returning to that focus on prediction.

Prediction in the Philosophy of Science

Popper (1959) famously saw making falsifiable predictions as a delimiter of science, while Whewell (1849/1969) thought that theories were proven by correct predictions of surprising new results. T. S. Kuhn (1996) saw scientific revolutions arising in part as the result of failed predictions (anomalies) and Duhem et al.

(1982) argued that the ability to predict novel phenomena could guide the choice between competing theories in the context of under-determination.

The precise role of prediction in science and understanding is definitely contested; for example, much ink has been spilled on the epistemic value of predicting versus accommodating evidence (Barnes, 2022), or on whether prediction is symmetric with explanation (Hanson, 1959; Rescher, 1958), but there is no shortage of voices hailing its importance and value. Strong arguments have been made that predictions can fortify theory, e.g. Whewell's and Duhem's, but also Douglas (2009), who puts it as "it is the ability of an explanation to generate new predictions, which then serve as a check on the explanation, that improves the accuracy of our scientific explanations". More importantly for us, prediction has been considered a goal of science in and of itself, as a pragmatic tool that allows planning and acting in the world. Hofstadter (1951) put it next to explanation as a central goal of science, and Longino (2002) went a step further, saying that "The purpose of scientific inquiry is not only to describe and catalogue, or even explain, ... but to facilitate prediction, intervention, control, or other forms of action on and among the objects in nature", which echos the foundational work of Hempel (1942, p. 38).

For our argument we do not need to believe in the value of prediction, or to claim it as the primary goal of science. We need only to understand that prediction is important and that it motivates the construction of knowledge in a general sense. We tend to look for explanations that can help us predict. This forms the basis for our conviction that a focus on explanations that can actually make predictions can help us improve the science of evaluation and our understanding of AI systems. By thinking about what sort of predictions we already have, which ones we need, and how we can make them, we can guide the search towards the explanations that would provide them.

Prediction in Evaluation

Another reason to focus on predictions is that evaluation is a prediction problem in its own right. By measuring model performance on a test set, we hope to predict its performance in the deployment setting. By doing error analysis we hope to tell which kind of input will induce which kind of failure (e.g. She et al. (2023)). We evaluate learning algorithms on multiple different scenarios in order to anticipate which other scenarios they will work in. The distribution shift and refinement problems we have mentioned before are actual prediction problems, which become actual hindrances for real world operation. There is a plethora of applications for powerful predictive capability, a major and obvious one being

the ability to prevent failures before they happen. We will discuss many more in section 4.1.

The advance of general-purpose AI (GPAI) –systems that can be useful in a wide range of tasks– has also created far-reaching implications for the field of evaluation and its predictive purpose. For one, GPAI entails the disappearance of deployment performance as coherent target to predict: a single model is used for a thousand tasks, directly by users or after being fine-tuned for specific applications. What are evaluators supposed to evaluate? Everything, all at once? There is no single measure that can reflect performance over the myriad of applications GPAI systems are used for. There is no more distribution shift, because there is no more distribution. Secondly, and related, the scope of evaluation has increased tremendously, in that current large-scale models, with their billion parameters and terabytes of training data, are orders of magnitude more complex than for example, a linear regression over a 3-dimensional dataset. Building good predictive models of performance has equally turned into a more involved effort, as there is more to learn about what has been learned. As a result, users catastrophically mispredict LLM performance (Carlini, 2023) and over-rely on their correctness (Steyvers et al., 2024). Lastly, even the basic predictive accuracy of evaluation has taken a hit, since as a result of web-crawling and increasing dataset size, test contamination through the leaking of test data into the training set has become a common phenomenon (Balloccu, Schmidtová, Lango, & Dusek, 2024). The new challenges in AI evaluation have a distinct predictive flavour.

Evaluation having a predictive function is not just the case in artificial intelligence. In the fields of psychology, psychometrics, and animal cognition the concept of predictive validity is well established, and refers to the ability of a test or other measurement to predict a future outcome. It is widely used and applies to various common settings such as predicting employee performance with job interviews or predicting academic performance with college admission tests.

Prediction in Artificial Intelligence

It is then particularly strange that AI evaluation produces such weak predictions. In a various number of AI sub-fields –machine learning, deep learning, control–, prediction is the language of choice and predictive models are the topic of study! The role of prediction is also well debated in the philosophy of these related and foundational fields, especially statistics and machine learning. See for example Breiman's deeply influential discussion on the two cultures in statistical modelling (Breiman, 2001), from which we take away that "using complex predictors may be unpleasant, but the soundest path is to go for predictive accuracy first,

then try to understand why". Or see Shmueli (2010), who argues that "using predictive modelling ... can help uncover potential new causal mechanisms" and that "it is important to establish prediction as a necessary scientific endeavour beyond utility". In the philosophical discussion on accommodating versus predicting evidence, some arguments rely on the same principles that are at the heart of statistics, see e.g. Hitchcock and Sober's (2004) argument based on the notion of 'overfitting'.

For a discipline so focused on prediction, artificial intelligence evaluation has very little predictive power. It seems only logical then, and at least worth a try, to develop a framework that roots evaluation in those same predictive formalisms and terms.

Personal Motivation

More personally, someone once told me that you only need a list of reasons if you do not have a convincing one. So while I believe all these arguments and goals to be sound and influential to my thinking, my personal motivation has always been simple intellectual curiosity. I have only followed my hunch that, if we could anticipate the performance of cognitive machines, it would mean that we understand intelligence, artificial or not.

Research Hypothesis

The problems listed above make clear that in many settings, evaluation simply is prediction, or that the two are at least closely related. As arguments for a predictive lens on evaluation we have also brought forward the fact that prediction is the language of choice for machine learning, and thus uniquely fitting the situation, as well that prediction plays a foundational role in the philosophy of science, which motivates explicit attention.

Having sketched this background, we can now explicitly state our hypothesis:

> By formally framing evaluation as a prediction problem, we can (i) bring together disparate methods and applications of evaluation of artificial intelligence, (ii) increase evaluation's predictive power for concrete applications that require it, and (iii) remediate debilitating issues in the science of evaluation, both old and new.

Research Objectives

Our research objectives follow our hypothesis closely.

Objective 1: Develop a Formal Framework. Firstly, we aim to develop a formal framework that supports the analysis of evaluation as a prediction problem. It should allow expressing the problems of evaluation, e.g. distribution shift and the refinement problem.

Objective 2: Survey and Integrate the Literature. With an adequate formalism in hand, we must survey, integrate, and organise the techniques and applications of evaluation. We must find overarching properties and use those to create taxonomies, draw new connections, and synthesise the science of evaluation.

Objective 3: Improve a Concrete Application of Evaluation. The synthesis of the previous objective should lead to the identification of gaps in the literature and new avenues of improvement making evaluation more predictive. Those improvements or ideas should be demonstrated and empirically validated for at least one concrete practical application of predictive evaluation.

Objective 4: Generate Theoretical Insight. The same formalism and synthesis should also lead to deep insight into remediating at least one of the fundamental challenges in evaluation, e.g. distribution shift, the refinement problem, or contamination. While this is an abstract objective, success can be measured in the plausibility of uncovered research directions, or if practicalities allow, implementation and demonstration of new paradigms of evaluation.

Research Method

There are three parts to the research method. First is the theoretical, or at least philosophical treatment of constructing a reasoning framework for predictive evaluation. We will assume that a 'scoring function' is given, or in other words, that the behaviour of an AI system can be graded, rated or judged, and a numerical score can be assigned to the behaviour. It is these scores that will be the target for prediction, and we will not concern ourselves with whether the scores are a valid measure of quality or what the semantics are behind the score. The next section provides additional motivation. Further, a core aspect of the formalisation process is to look at the fields of machine learning and statistics, which are ripe

with predictive formalism to borrow from. To strengthen the framework, we have (i) sought to publish key ideas quickly as to gather feedback, e.g. Hernández-Orallo et al. (2022), and (ii) sought collaboration from a multitude of scientists from across a wide range of disciplines, e.g. in efforts such as Burnell, Schellaert, et al. (2023), Schellaert et al. (2023) or L. Zhou et al. (2023).

A second aspect of our research has been intensive literature analysis and integration. We have tried to connect disparate fields and techniques, moulding them to the same structure, both to draw broad connections and to test the boundaries of our framework. The methods we investigate do not limit themselves to the standard practices in artificial intelligence; we also look at techniques from psychometrics (Item Response Theory, section 4.2.2.2), animal cognition (Capability-Oriented Evaluation, section 4.2.5.4), and cognitive philosophy (e.g. cognitive extension, section 3.4); leading to diverse perspectives in fruitful cross-disciplinary investigations, e.g. Voudouris et al. (2023, 2024) and Schellaert, Hamon, et al. (2024). And while not a traditional part of scientific methodology, we have maintained a newsletter on advances in AI evaluation[1], which we believe to have fortified our literature analysis.

Third and final is the empirical validation and investigation, making up a significant portion of this research (Chapters 5 and 6). The methodology is as follows: (1) find an AI system of interest with certain evaluation pathologies, e.g. ChatGPT (Ouyang et al., 2022), (2) collect evaluation data for that system across a wide range tasks, constructing benchmarks if necessary, and (3) use that data to compare different predictive models of performance, investigating how well they predict scores, and what factors influence it.

As subjects of study, we give all our attention to Large Language Models (LLMs). While LLMs were barely emerging when this research started, they have quickly grown to be a dominant topic in global AI discourse, industry, and research. Their impact has been immense, with apps like ChatGPT amassing millions of daily active users, billions of dollars in investment and funding, and serious policy initiatives aiming to regulate GPAI around the globe, e.g. the AI Act (European Commission, 2024) or California's SB-1047 (State of California, n.d.).

More importantly, as a subject of study, LLMs and their evaluation provide us with all pathologies and challenges of interest: distribution shifts, disparate evaluation goals, general-purpose utility, test contamination, high sensitivity in scores... Like a rainforest for an ornithologist, large language models provide richness, depth, and diversity to an evaluation researcher. Therefore, they are the perfect playground to put our theory and ideas into practice.

[1] https://aievaluation.substack.com/

Scope

With regards to Objectives 3 and 4, this thesis will not be exhaustive or systematic, in the sense that these objectives are broad and flexible. If our framework (Objective 1), and our synthesis (Objective 2) are any good –and here do we try to be as comprehensive and structured as possible–, a rich well of new research directions and ideas should spring from it. Our follow-up work is therefore necessarily explorative. We have picked up the theory and experiments that were most prominently important, the most clear, and that best fit the reality of the day. For example, we believe the study on human score prediction in Chapter 6 to be particularly impactful.

We also limit ourselves (i) to predicting only the performance of AI systems, and not its general behaviour or outputs (appendix B.1 explains why), (ii) by not considering the output of the AI system as an input to performance prediction (see B.2), and (iii) by focusing on prediction of instance-level scores instead of aggregate performance, although there is plenty of nuance and complexity to that statement (see appendix B.3).

We have also focused on quantitative analysis and predictions of scores. With the rising prominence of AI in our daily lives, evaluation has increasingly begun to encompass the analysis and prediction of social, economical, and cultural impact. We briefly touch upon it in section 3.4, but pay no further attention. As Meehl posits (Meehl, 1954, 1967), the "hard" sciences have the advantage of being able to formulate precise and falsifiable predictions, and we have thus aimed to reap that benefit.

Sometimes, the language might be over-abstracted and speak about ML/AI in general, while our mind's eye was on e.g. a classic tabular data supervised learning setting or large language models instead, but not necessarily both, or not necessarily including e.g. reinforcement learning or purely unsupervised problems. Much of the formalism and discussion should generalise broadly, but it is entirely possible that some thought patterns make more sense in a reduced scope. As it is notoriously hard to define what "AI" or an "AI system" is, we will refrain to use or create such a definition, and instead count on the reader to imagine and delimit appropriate examples themselves.

And while we have littered our reasoning with the claim that 'evaluation is prediction', this is a totum pro parte. Evaluation occupies itself not only with predicting the quality of system behaviour, it also determines that quality in the first place. As any teacher can tell you: it can be hard to find a consistent and honest scoring system, especially for open ended questions. Scoring (or grading, judging,

verifying, or rating) requires knowing the task at hand, and it can be a labour-intensive and costly process, no less so in the field of AI. Grading AI systems is a research topic on itself (Gehrmann et al., 2023; Zheng et al., 2023), as we said in the research method section, we will consider it a given. This is because we believe grading to be hard precisely because of the predictive function of evaluation. Apart from pure competition such as the Olympic Games, a measurement of quality is ultimately always turned into a statement about something other than the behaviour just observed: it is a reflection of what the system can do, and it attaches implications about *future* behaviour of the system. A student's test must be graded, but it is ultimately a reflection on the student's understanding of the material and their ability to successfully complete similar challenges. At that point, we would be back at square one, as the hart of the matter is again prediction.

Although we provide a introduction to the most pertinent topics in Chapter 2, we have assumed knowledge of the fundamentals of statistics and the high-level concepts of machine learning. It also helps to know what a large language model (LLM) is and how their derivative chatbots and virtual assistants look like in the year 2024.

Chapter Outline

The next chapter, Chapter 2, provides, among other things, an introduction to uncertainty estimation (section 2.2), evaluation methodologies (section 2.4, and the current challenges of evaluation (section 2.5). After that background, the dissertation has has four main chapters. The first one, Chapter 3, frames the evaluation of artificial intelligence as a prediction problem. It lays conceptual and formal groundwork (section 3.1), and since predictions can be wrong, follows up with a discussion on "evaluating evaluation" (section 3.2). We then provide further analysis on the relation between prediction and predictability (section 3.3), and how users can be considered as an additional dimension to score prediction (section 3.4). We finish with a discussion on how the challenges of evaluation fit in our framework (section 3.5). This chapter is supported by an FAQ in appendix B, which provides extended motivation for some of the core design choices made so far.

With those fundamentals in place, we can organise the practice of AI evaluation, structuring the plentiful inter-connections in Chapter 4, both in terms of applications (section 4.1) and in terms of methods (section 4.2). We complete

that chapter with a summary overview of common properties of score predictors (section 4.3).

While Chapters 3 and 4 are theory, philosophy, and connecting far edges of literature, Chapters 5 and 6 will instead be pure empiricism. Particularly, we will look at large language models (LLMs), and how their scores can be predicted by both (smaller) language models, and by human models of performance.

In terms of predicting language model scores with language models, in Chapter 5 our focus will be on our newly developed assessor models (section 4.2.5.3) with BERT-style architectures, investigating the general predictability of LLM subject scores and the factors that influence it, e.g. the use of multi-subject data, multi-task data, assessor scale, and subject scale, etc.

In terms of humans doing the score prediction, in Chapter 6 we will analyse the predictive power of human-derived notions of difficulty, how prompt-sensitivity influences predictability, and the need for score prediction due to a lack of subject self-assessment to rely on. These high-level concepts are interpretable and accessible to users, and are useful across tasks. We finish with an extensive human study showing how a priori performance expectation management integrates with a posteriori supervision of the generated outputs. This chapter is supported by additional analysis and background in appendix C.

Appendix A lists all the publications that have been integrated in this dissertation.

How To Read

The introduction is needed to understand why this thesis is being written, but if you are convinced already, there are only two sections that you need to understand to read everything else. Those are section 3.1 on the Score Prediction Problem, which sets out our basic formalism, and section 4.3, which defines the vocabulary and concepts that are needed to differentiate between a plethora of closely related methods, use cases and experiments throughout the rest of this work.

Apart from those two sections, you are free to pick and choose parts in whatever order you see fit and follow backreferences if needed, although finishing Chapter 3 first is recommended. If you were not convinced by the motivation in the introduction, a sneak peak of section 4.1 on applicability might help. Chapter 3 is likely to be the most timeless, but is also more abstract. Chapters 5 and 6 contain more concrete and immediately useful findings.

AI	Artificial Intelligence
ARC	Accuracy Reject Curve
AUROC	Area Under the Receiver Operating Curve
BS	Brier Score
ERC	Error Reject Curve
FAQ	Frequently Asked Questions
GAN	Generative Adversarial Network
GPAI	General-Purpose Artificial Intelligence
i.i.d.	Independently and Identically Distributed
IRT	Item Response Theory
kNN	k-Nearest Neighbour
LLM	Large Language Model
LM	Language Model
MAE	Mean Absolute Error
ML	Machine Learning
NAS	Neural Architecture Search
NLP	Natural Language Processing
NN	Neural Network
OOD	Out of Distribution/Domain
PPR	Perfect Predictability Region
ROC	Receiver Operating Curve
RL	Reinforcement Learning
SVM	Support Vector Machine

Table 1.1: Overview of acronyms.

z	Lowercase letters denote a variable.
$\mathbf{z}$	Bold lowercase letters denote a variable where we want to emphasise the variable is vector-valued.
Z	Uppercase letters denote either a random variable, a set, or a matrix, which will be clear from context.
$\mathcal{Z}$	Calligraphed letters denote spaces.
$\mathbb{P}_{\mathcal{Z}}$	The space of probability distributions over the space $\mathcal{Z}$, either discrete or continuous.
x	An instance of a problem.
$\mathbf{x}$	A representation of instance x as a vector of features.
$\mathcal{X}$	The space of possible instances.
X	A random variable of instances.
m	An AI system under evaluation, also referred to as a *subject*.
$\mathbf{m}$	A representation of subject $\mathbf{m}$ as a vector of features.
$\mathcal{M}$	The space of possible subjects.
y	A behaviour of a subject.
$\mathcal{Y}$	The space of possible subject behaviours.
Y	A random variable of subject behaviours.
$m(\hat{y}\|\mathbf{x})$	Distribution over behaviours $\hat{y}$ for subject m given instance $\mathbf{x}$.
$m(\mathbf{x})$	A concrete sampled behaviour of a subject m for instance $\mathbf{x}$.
r	A score
$\mathcal{R}$	The space of possible scores.
R	A random variable of scores.
$\dot{y}(y, \mathbf{x})$	The score produced by scoring function $\dot{y}$ for a behaviour y on a particular instance $\mathbf{x}$.
$\dot{R}$	A collection of evaluation results $\{\langle \mathbf{m}_1, \mathbf{x}_1, r_1 \rangle, ...\}$.
s	A score predictor.
$s(\hat{r}\|\mathbf{m}, \mathbf{x})$	Distribution over scores $\hat{r}$ given subject $\mathbf{m}$ and instance $\mathbf{x}$.
$s(\mathbf{m}, \mathbf{x})$	A concrete sampled scores given subject $\mathbf{m}$ and instance $\mathbf{x}$.
$\mathcal{L}$	A loss function $\mathbb{P}_{\mathcal{Y}} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$.
$\mathcal{D}$	A divergence $\mathbb{P}_{\mathcal{Y}} \times \mathbb{P}_{\mathcal{Y}} \to \mathbb{R}_{\geq 0}$.
$\epsilon_D[f]$	The empirical risk for a predictive function f on dataset D.
ϵ	Noise

Table 1.2: Overview of notation.

Chapter 2

Background

In this chapter we will briefly discuss some topics that are important and will often return, but are not part of our own research. These include the concept of general-purpose AI (GPAI), the principles and techniques for estimating uncertainty in AI systems, and a variety of evaluation methods, including aggregate metrics, error analysis, and item response theory. For readers that are familiar with the discussed topics, the corresponding sections can be skimmed or skipped, although we recommend reading sections 2.5 and 2.6, as these provide a more unique and pertinent perspective on the challenges of evaluation that we will try to tackle.

2.1 General-Purpose AI

Economists use the term general-purpose technology to refer to a technology that can affect an entire economy at global or national level (Lipsey et al., 2006). Examples are the steam engine, electricity, or the computer. Artificial intelligence itself is general-purpose, but the criteria included in Lipsey et al. (2006), apart from "has many different uses" and some others, include "being a single, recognisable generic technology", which might not apply to artificial intelligence as a whole. Perhaps relatedly, the term general-purpose artificial intelligence (GPAI) has only really come in use when a single trained AI system could be used for "many different purposes".

Eloundou et al. (2023) has argued that Large Language Models (LLMs) are such general purpose technologies. Broadly speaking, LLMs are AI systems, usually based on the Transformer architecture (Vaswani et al., 2017), that are trained in a self-supervised manner from large internet-based training corpora to predict the next most likely character given a previous sequence of characters. They model $p(c_t|c_{t-1}, ..., c_0)$, where c represents a character (or group thereof, often called a token), and t represents the order of those characters in a text. They are the large scale implementation of the idea of language models from Shannon's (1948) foundational work on communication. Their general utility is derived from the fact that, seemingly, after a stated question often comes a correct answer. With a language model in hand, we can repeatedly sample continuation characters –at each step sampling c_{t+1} in autoregressive fashion, i.e. based on the given question and already sampled tokens– to produce that answer.

While these technical ideas are still central, the term GPAI is also used for a wide range of derivative and similar systems that can be used for more applications than can be enumerated. A single instantiation of such a system can, for example, do question-answering, translation, code generation and text transformation at the same time. Some systems are multi-modal, and can reason over images as well, or generate them based on an input prompt. The earliest GPAI systems could be considered to be BERT-style pre-trained language models (Devlin et al., 2019), but these still required additional training –referred to as fine-tuning– to adapt them to specific use cases, and were originally described as foundation models rather than general-purpose AI (Bommasani et al., 2021).

Chatbots such as ChatGPT (Ouyang et al., 2022) on the other hand, are not just adaptable, these models are general in a way that allows them to tackle a wide range of tasks without additional training, as long as the tasks can be framed in natural language (Chung et al., 2022). In addition to self-supervised learning, more recently they have also been trained with large collections of instruction-output pairs in a supervised manner, and with reinforcement learning from human feedback, both to make the system more instructable in regular natural language (Ouyang et al., 2022). As we will see, this flexibility and generality will make their evaluation harder. Other important properties are that they are often used directly by end users, experts and non-experts alike, for various everyday tasks.

2.2 Uncertainty Estimation

As AI systems are increasingly deployed in high-stakes decision-making scenarios such as healthcare and finance, it has become crucial to quantify the uncertainty associated with their actions, or in the case of machine learning models, their predictions. In this context, uncertainty estimation refers to the process of assigning a measure of confidence to the predictions made by a predictive model. This is essential for several reasons: (1) to avoid overconfident predictions that may lead to poor decision-making down the line, (2) to identify situations where the model is likely to make mistakes, and (3) to provide a more comprehensive understanding of the model's limitations.

In this section, we will discuss probabilistic models, the concepts of calibration, the difference between aleatoric and epistemic uncertainty, and how uncertainty is represented in language models. We will also discuss the broader implications of effective uncertainty estimation for improving the reliability and transparency of AI systems.

2.2.1 Probabilistic Models

Probabilistic models are a natural fit for uncertainty estimation because they provide a mathematical framework for representing and manipulating uncertainty. These models learn to predict a probability distribution over the possible outcomes, rather than a single point estimate. Instead of functions $\mathcal{X} \to \mathcal{Y}$, they are functions $\mathcal{X} \to \mathbb{P}_{\mathcal{Y}}$, where $\mathbb{P}$ represents the space of probability distributions over $\mathcal{Y}$, either discrete or continuous.

Some machine learning techniques are inherently probabilistic, e.g. logistic classifiers, neural network classifiers ending with a soft-max layer, or Gaussian processes. Others, e.g. linear regression or support vector machines, are less so. But for virtually every machine learning method there are techniques for extracting some notion of decision uncertainty. Common and general ones include ensembling (Kuncheva, 2014) and Platt scaling (Platt, 1999). We refer to the books from Murphy (2022) and Murphy (2023) for an extensive overview of probabilistic modelling. We will often refer to probabilistic models as doing *self-assessment*.

Probabilities and Language Models

As mentioned above, language models are also naturally probabilistic, as they assign probabilities $p(c_t|c_{t-1}, ..., c_0)$ to the next most likely character c_t given a sentence $q = c_0...c_{t-1}$. Autoregressive sampling then produces a continuation $c_t...c_n$, which acts as an answer to the question or task represented by q. We can then calculate the confidence in that answer as $\prod_{i=t}^{n} p(c_i|c_{i-1}...c_0)$.

One problem with this approach is that natural language already has terms to express uncertainty, e.g. 'maybe', 'probably', or 'certainly', and users expect uncertainty to be expressed in the text (Lin et al., 2022; Mielke et al., 2022). The other, more fundamental issue is that many different continuations can constitute a correct answer, and probabilities will be spread over all of them, providing an inaccurate estimate of uncertainty (see also the notion of calibration in the section below). The problem worsens as the outputs become longer, forcing techniques such as length normalisation (Murray & Chiang, 2018) to salvage the problem. We refer to Baan et al. (2023) for an overview of uncertainty in LLMs.

2.2.2 Calibration

Calibration refers to the notion that estimated probabilities should reflect the true likelihood of the event. For example, if a model predicts a probability of 0.8 for a particular class, it should be correct approximately 80% of the time. Calibration is essential for building trust in machine learning models, since if the model is not calibrated, i.e. a 10% chance of disease is not actually a 10% chance, we cannot rely on its prediction confidence in our decision making.

There are several notions and definitions for calibration. For classification, the most stringent definition is the following (Filho et al., 2021):

> A probabilistic classifier $f : X \to \mathcal{P}_Y$ is multiclass-calibrated if for any prediction vector $s = [s_1, \ldots, s_K] \in \mathcal{P}_Y$, the proportions of classes among all possible instances on the random variable X getting the same prediction $f(X) = s$ are equal to the prediction vector s.
>
> $$P(Y = i \mid m(X) = s) = s_i \quad \text{for } i = 1, \ldots, K. \tag{2.1}$$

An important concept for calibration is proper scoring rules (Gneiting & Raftery, 2007). A score (or error) assigned to a probabilistic prediction to reflect its correctness is only proper if it creates an incentive for the model to reflect true probabilities. This is not always so, e.g. the absolute error $|y - p(y)|$ creates

an incentive to report extreme probabilities. To illustrate this, consider a binary classification scenario where the true underlying probability of the positive class is 0.9; if a (binary) predictor would always report the true $p(1) = 0.9$, its average error would be $90\% \cdot 0.1 + 10\% \cdot 0.9 = 0.18$. However, if the model instead reports an extreme probability, such as $p(1) = 1$, its average error would only be $90\% \cdot 0 + 10\% \cdot 0.9 = 0.09$. Examples of proper scoring rules include the log-loss, and Brier score (Brier, 1950).

Much of the focus in calibration is on classification tasks, but methods for regression also exist. For a comprehensive overview, Filho et al. (2021) is a great resource for classifier calibration, and H. Song et al. (2019) fills that function for regression calibration.

The concept of calibration in general is the subject of intensive research, see e.g. Bella et al. (2010), Y. Chen et al. (2022), Dimitriadis et al. (2021), Guo et al. (2017), Kumar (2022), Nixon et al. (2019), and Platt (1999) or Steyvers et al. (2024).

2.2.3 Aleatoric vs. Epistemic Uncertainty

Uncertainty can be broadly categorised into two types: aleatoric and epistemic (Hüllermeier & Waegeman, 2021). Aleatoric uncertainty, also known as irreducible uncertainty or ambiguity uncertainty, arises from the inherent randomness or noise in the data and observations. This type of uncertainty is inherent to the problem and cannot be reduced by collecting more data or improving the model. A typical example is that of the fair coin toss: no matter how many coin tosses you observe, you won't be able to predict the result of the next toss with more than 50% accuracy (on average)[1].

Epistemic uncertainty, on the other hand, arises from the limitations of the model and the data used to train it. This type of uncertainty can be reduced by collecting more data, improving the model architecture, or using more advanced training techniques. Understanding the distinction between aleatoric and epistemic uncertainty is essential for developing effective uncertainty estimation methods, as it allows for targeted approaches to address each type of uncertainty.

[1]This concept is often abused and used to mean that uncertainty cannot be reduced with the *type* of data that is collected, i.e. the variables that are observed. Even for a fair coin toss, if enough is known about initial conditions of the coin, the force applied, launch angle etc, the landing face of the toss can be predicted. We usually assume we can not observe these variables, and then the uncertainty of the coin toss is indeed irreducible.

Disentangling the two types robustly is a topic of intense research, see e.g. Hüllermeier and Waegeman (2021) and Kendall and Gal (2017), and for the purpose of predicting performance, there are different methods for each, see e.g. Markou and Singh (2003) for novelty detectors, which focus on epistemic uncertainty.

2.3 Distribution Shift

As we have mentioned earlier, distribution shift generally means that a particular setting –a data-generating process to be more precise– changes over time.

In supervised learning, we typically have a set of training samples from a source distribution $p(y, x)$ available to us. We use these samples to fit a model $p(y|x)$. At deployment time, we encounter data from a distribution $q(y, x)$. When $p \neq q$, we say that distribution shift has occurred.

Let us assume a causal relationship, i.e. that x causes y. An example task is house price modelling, with x being features of the house, e.g. the number of rooms, the presence of a garden, etc., and with y being the corresponding price, which we aim to estimate with machine learning. In this scenario, we can factor the joint distribution into prior $p(x)$ and discriminative model $p(y|x)$, i.e. $p(y, x) = p(x)p(y|x)$, From this we can derive two different types of distribution shift. One is called covariate shift, where $p(x)$ changes, but $p(y|x)$ remains the same, e.g. we start seeing more flats than villas, but the pricing relationship stays the same. The other is called concept drift, and occurs when $p(y|x)$ changes, i.e. for the same house, prices would now be different. Concept drift and covariate shift can occur at the same time.

We could also assume an anti-causal relationship, i.e. y causing x, for example disease y causing symptoms x. Many tasks in machine learning are anti-causal, but we will refer to the excellent Murphy (2023, Sec. 19) for further exposition.

2.4 Introduction to Evaluation Methods

Before we present our framework (Chapter 3) and organisation (Chapter 4) for the science of evaluation, we should warm up a little. The next few sections will explain some concrete evaluation methods, some of them potentially unfamiliar, and provide something concrete to hold on to when the conversation gets more abstract later on.

2.4.1 Aggregate Metrics on a Hold-Out Test Set

At the core of most current evaluation is the estimation of the generalisation error of a model, also known as the infinite test set error or the risk. It is typically formulated as the expected value of a loss function over some assumed distribution of inputs X and prediction targets Y. Since we realistically work with a finite amount of data, we have to approximate and measure the empirical risk instead, i.e. the loss over a finite amount of samples. In other words, given a set of samples, we (i) collect model outputs, (ii) calculate the loss and (iii) take the average. This is then seen as the estimation of the generalisation error.

It is well understood that we should not estimate the empirical risk using the same data that was used to train the model. Depending on its learning capacity, a model can perfectly fit its training data (and in many different ways), leading to overfitting: a seemingly accurate model that does not generalise to new data points. To produce a better risk, techniques such as cross-validation and hold-out test sets are used (Browne, 2000).

The importance of prediction in evaluation should be clear: seeing estimation as prediction, it is at the core of even the simplest evaluation procedure. And the notion of simply measuring performance on a set of test cases assumed to be representative is of high interest. It is the dominant practice, and will act as a sort of baseline reference against which other evaluation methods can be compared. This is because, apart from measuring generalisation error based on the loss function, there are various other metrics that measure something different (but often related), and all together, these metrics cover much of the utility of evaluation. There is e.g. accuracy, which is based on the binary correctness of a classification, or the R^2 metric for regression, which measures the proportion of variation in the target Y that can be explained by the model. There are thousands of domain-specific metrics (e.g. BLEU (Papineni et al., 2002)), ones dealing with data-imbalance (e.g. AUROC (Fawcett, 2006)), or those measuring calibration (as seen in section 2.2.2 above).

2.4.2 Error Analysis

Aggregate metrics already cover a lot of ground, but they can be further amended by *error analysis*, a very broad idea which roughly entails analysing when, how, and why models fail. Due to the general vagueness, the literature on error-analysis seems only sparsely connected, and techniques are often tailored to specific settings, e.g. specific modalities, ML techniques, or application domains. Nonetheless, some general patterns can be observed.

Following Wu et al. (2019), two common forms of error analysis are '(1) data grouping, where aggregate metrics are computed for particular slices of interest (e.g., accuracy over question types in machine comprehension, or over queries containing different types of negation) and (2) counterfactual error analysis, where one modifies the input data and assesses if expectations are met still met, for example adding irrelevant data and seeing if new errors are introduced.' In NLP, examples of data grouping are X. Liu et al. (2018) and She et al. (2023), while Jia and Liang (2017) and Ribeiro et al. (2018) perform counterfactual analysis. S. Lee et al. (2022) describes a data grouping example for vision models, and computer vision in general is ripe for counterfactual error analysis as a byproduct of the significant interest in models that are robust against adversarial image generation (Carlini & Wagner, 2017). Definitely not every technique falls neatly into these two categories, see e.g. Khanna et al. (2022) for error analysis in reinforcement learning.

In general, error-analysis is a manual or semi-automated process, focused on human understanding.

> **What's the matter?** Error analysis captures much of what is really important in evaluation: understanding how the problem and the model relate to the measured performance. It is not done often enough, and in Burnell, Schellaert, et al. (2023) we argue –with a host of renowned scientists– to do more of it. But it is not a *grounded* practice. There is very little theoretic or general scientific backing. Pragmatism is the central driving force. For a development looking to analyse their model before deploying it, this does not need to be a problem, but if we want to understand, generalise, and automate the effectiveness of error analysis, we need a more structured approach, and at least a conceptual framework that can tie the field together.

2.4.3 Learning Curve Analysis

The methods we have discussed so far have focused on evaluating the model with the parameters set and learned, ready to potentially deploy. But we can also evaluate the 'design of the model'; in the context of neural networks for example, this would be its architecture, the specific choice of activation functions, the use of convolutions, transformer blocks etc.

Scaling analysis is particularly interested in how good the model will get as we train it for longer, e.g. using more epochs, or if we make it bigger, e.g. adding more layers, or if we just add more data. The literature has consistently found that there

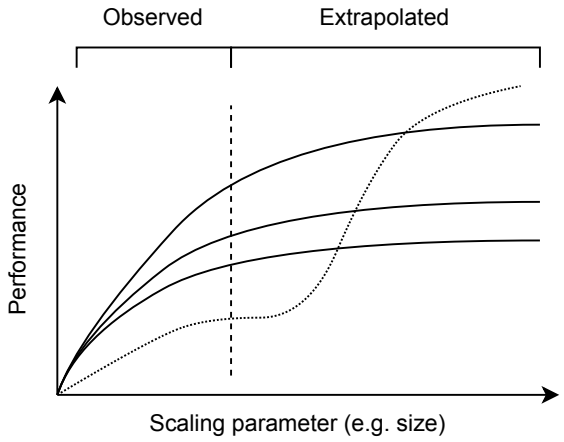


Figure 2.1: Learning curve example. The full lines are the result of plotting the performance of the different model designs against its scaling parameter, e.g. number of layers, size of the training dataset, epochs of training. We see consistent behaviour. By observing performance up to a certain scale, we can select an optimal design for a larger version, because we can extrapolate the performance to that scale. The dotted line represents a model design that would break that assumption, but we generally do not observe those unpredictable learning curves.

are predictable patterns between the performance of the smaller/earlier versions of the model, and the larger/later versions. Figure 2.1 shows an illustration.

The predictable patterns of performance allow us to do model selection before training all of the larger variants, which might be expensive, allowing us to pick a single model to invest resources into. Learning curve analysis will come back in section 4.1.1.2 and section 4.2.3.1.

2.4.4 Item Response Theory

Item Response Theory (de Ayala, 2009) is a framework from psychometrics oriented around question difficulty. It assumes that different questions have different difficulty levels, and that different subjects –test-takers either human or machine– have different capability levels. Both the difficulty of the question and the capability of the subject are latent variables: they cannot be observed, and must instead be deduced based on observed scores. More so, for a given set of questions, IRT assumes the difficulty and capability are uni-dimensional, in that there is a single latent factor for which questions can be more difficult, and in which subjects can

be more capable. A subject that is more capable should have a higher chance of answering the question correctly.

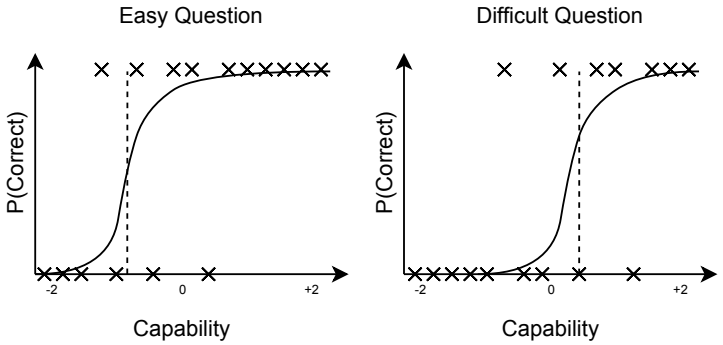


Figure 2.2: Two different Item Response Curves. The question on the left is more easy than the question on the right, as subjects with a lower capability have more chance of answering the question correctly. The difficulty is determined by the location of the slope of the logistic curve.

IRT assumes that for each question, we can deduce an 'Item Response Curve', a mapping from subject capability to the chance of answering correctly on that question. The Item Response Curve follows a particular parametrised function $f(\theta; d)$, e.g. a logistic function, where θ represents capability, and the parameters d are tied to the specific question.

Both θ and d are fitted from empirical responses, i.e. scores gathered for a collection of subjects and a collection of questions. In that way, IRT is populational, it cannot assess the capability of a subject without other subjects. Item Response Theory will be further discussed in section 4.2.2.2.

2.4.5 Separated Rejectors

Machine learning with a reject option (Hendrickx et al., 2024) refers to the notion that, given some expression of uncertainty about a model's output, we can reject using that output if the uncertainty about the correctness is too high.

Hendrickx et al. (2024, Sec. 4) describes 'separated rejectors', which are functions $\text{rej}_f : \mathcal{X} \to \mathbb{R}$, that assess whether a certain input x needs to be rejected for model f, without relying on f or its expression of uncertainty. The composite system combining f and rej_f looks like this:

$$f'(x) = \begin{cases} \text{REJECT} & \text{if } \text{rej}_f(x) < \tau \\ f(x) & \text{otherwise} \end{cases},$$

with τ some threshold set to balance the number of rejections with the desired level of uncertainty.

This is not usually considered as 'evaluation', but r_f is explicitly a model of performance of f. They assess when a certain input x is too difficult –usually because it is too novel– for the model of interest f. This line of thinking will be central in our analysis.

2.5 Prediction in Evaluation: Classic Challenges

If we want to talk about improving evaluation, we should talk about what there is to be improved. We will start with old issues that have plagued the field from early on.

2.5.1 Test Adaptation

Test adaptation is a particular type of overfitting. The term is used to refer to the fact that a model overfits the test set, even though test instances were not part of its training set. This happens because the optimisation process of a learning algorithm is actually nested within a higher-level optimisation: that of the model developers' selection of optimal hyperparameters and architectures, often colloquially called 'grad-student descent'. The developers are repeatedly exposed to the results of the test set and continuously tweak the model design to improve the score. If this is done enough times, there is a good chance that that result is a fluke: the model generalises from training to testing, because a design that does so was chosen by chance, but the model still does not generalise to new data.

For this reason, developers often use two hold-out sets: a validation set, used for second-level optimisation, and then an actual test set, to which exposure should be limited. The problem however returns with public benchmarks: teams from across the globe look at each other's techniques and results, copy what works best, and optimise it some more, until it stops being a measure of generalisation. As the famous adage known as Goodhart's law goes 'when a measure becomes a target, it ceases to be a good measure'. Gencoglu et al. (2019) provides further discussion on test adaptation.

2.5.2 Distribution Shift

In section 2.3 we explained how distribution shift affects machine learning models. Now, distribution shift affects the evaluation of machine learning models just the same. If either the covariates $p(x)$ or the discriminative relation $p(y|x)$ changes, the accuracy of the estimated error will be affected too. The model $\hat{m}(y|x)$ might be more accurate for some x than for others, and if we see a relative increase in the occurrence of these more difficult instances, performance will be lower.

The standard evaluation procedure of calculating aggregate metrics on a hold-out set therefore assumes distributional equality. Framed in terms of prediction, the claims made by these evaluation procedures are very narrow in scope, since no estimates are made of performance for any different distribution.

2.5.3 Refinement

Another problem we have mentioned earlier is the lack of granularity in performance prediction. The only artefact of most evaluation procedures is the calculation of some aggregate metric, along the lines of "model X is Y% accurate on dataset Z". Because these simple aggregate metrics are so ingrained, more granular predictions, e.g. "model X has Y% chance of getting the answer to particular question Z correctly", almost do not feel like evaluation anymore. Do people really care about that? Burnell, Schellaert, et al. (2023) argue they do, and there are plenty of use cases for this granularity (see section 4.1). A more important reason for framing this as an evaluation task is that the ability to make these predictions is often a function of empirical performance results, as is the case with most evaluation procedures. However, simple aggregate metrics do not provide the needed predictive power. They are well calibrated, but have no refinement or precision. They are like constant functions, simply outputting the mean score or chance of correctness, regardless of the input.

2.5.4 Lack of Prediction Target

The standard evaluation explained in section 2.4.1 works quite well if you can get your hands on a representative test set. Often, especially in the academic world, we are not evaluating a model for a specific use case, but the quality of a new model architecture or learning algorithm for general use[2]. In this case, what

[2]We ourselves will typically use model to mean a specific trained model, and use architecture for the 'untrained' design of the model. If we say system we also mean a trained model, typically embedded in some use case or higher-level software (e.g. a chatbot interface).

should the test set be? This is decided by downstream model developers and their specific settings.

As a work-around, the scientific practice is to test the learning algorithm on many test datasets at the same time, with the assumption that an improvement in performance over previous algorithms on the majority of datasets entails an improvement on the majority of realistic uses[3]. To follow the traditional archery analogy popularly used to explain the concept of bias and variance (see fig. 2.3), it is difficult to predict performance for learning algorithms, because there is no real target to shoot at, or at least we cannot see it, so we shoot at fifteen different ones instead.

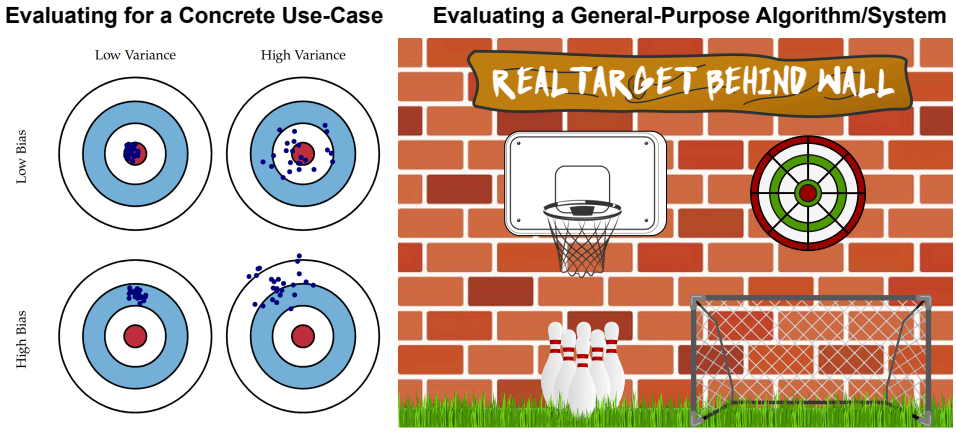


Figure 2.3: (left) Figure popularly used to explain bias and variance of statistical estimators. With a representative test set, we can compare model variance, bias, and the expected generalisation error. From Fortmann-Roe (2012). (right) Estimating generalisation error for learning algorithms or GPAI systems. Prediction is hard, because the target (the data distribution) cannot be observed at all. Instead we shoot at other targets, in the hope performance will translate.

[3]Publishing incentives however facilitate only reporting the performance on the datasets for which the hypothesis holds.

2.6 Prediction in Evaluation: New Challenges

We also want to highlight some issues that have popped up or became more prominent as the field has evolved.

2.6.1 Lack of the Prediction Target (Trained Models)

The advance of general-purpose AI (GPAI) –systems that can be useful in a wide range of tasks– has far-reaching implications for the field of evaluation: GPAI entails the disappearance of deployment performance as a prediction target even for a single concrete trained system. What are evaluators supposed to evaluate? A single model is used for a thousand tasks, directly by users or after being fine-tuned for specific applications. There is no more distribution shift, because there is no reference distribution. The closest thing to an empirical distribution might be found in 'arenas', free-for-all collections of user ratings of GPAI behaviour (Chiang et al., 2024).

2.6.2 Enormous Scope

Related to the above, building (mental) models of AI performance has also become more difficult as the models and training datasets have become several orders of magnitude larger. The behaviour of some current models exhibit, notably large language models (LLMs), is much more complex than for example, a linear regression over a 3-dimensional dataset. In a small regression scenario, one can reason about outliers and interpret the metrics (e.g. MAE). But what is an outlier for an LLM and what does a 0.23 perplexity on a trillion token dataset entail? New behaviours emerge through the interaction of a billion parameters and data points. We cannot grasp the amount of data used to fit such models, let alone anticipate which question it could reasonably answer correctly. Users catastrophically mispredict LLM performance (Carlini, 2023) and over-rely on their correctness (Steyvers et al., 2024). The scope of evaluation has increased, there is more to learn about what our models have learned, and more powerful methods are needed to model our models and their performance.

2.6.3 Contamination

As a result of web-crawling, increasing dataset size, and the general opaqueness of the training data, the issue of test contamination has become prominent for GPAI systems (Balloccu, Schmidtová, Lango, & Dusek, 2024). Contamination happens when models are evaluated with problems that are also present in the training data, either exactly, or with great similarity. We have already seen that this sort of *accommodation* of data –i.e. overfitting– hinders the generalisation, and thus the predictive utility, of evaluation results. To create fair evaluations, Dominguez-Olmedo et al. (2024) even suggests to let all participant models explicitly train on the test set equally. But as we will see later, this only creates fair rankings, not useful evaluations.

Closing Notes

The notion of evaluation measures as a statistical estimator is of course not new. The bias and variance of various evaluation procedures has been researched for a long time, and it is core to the science of evaluation (Browne, 2000). But an estimator as such only provides a single prediction: for a sample of instances from a similar distribution, performance will be about the same. It tells us little about the expected score for specific instances, what the expected performance is when distribution shift occurs, or how to deal with the other problems listed above. For that, we will need score prediction with a proper predictive perspective on evaluation.

Chapter 3

Evaluation as Prediction

In this chapter, we introduce the problem of score prediction in the context of AI system evaluation. We aim to formalise evaluation as a predictive modelling task similar to traditional supervised machine learning. We begin by defining the necessary components of learning, such as the input space, the target space, and the evaluation records. From there, we will discuss the coupling of predictive power, predictability, and performance, introducing the concept of subject predictability and exploring how it can be balanced with model performance. We will expand the scope of score prediction to include user inputs, recognising that the utility of AI systems is often determined by the interaction between the system and the user. Finally, we will revisit the challenges of evaluation in the context of our framework, tackling the various challenges associated with accurate score prediction, and summarising the upsides and limitations.

3.1 The Score Prediction Problem

In the traditional supervised machine learning setting, a model m is a mapping $\mathcal{X} \to \mathcal{Y}$, or if probabilistic, a mapping $\mathcal{X} \to \mathbb{P}_{\mathcal{Y}}$, where $\mathbb{P}_{\mathcal{Y}}$ refers to the space of probability distributions over $\mathcal{Y}$, either discrete or continuous. As an example setting we can consider the classification of sentiment in textual movie reviews. Here, our input space $\mathcal{X}$ would be the set of possible reviews, and the set of

sentiments we wish to classify the review with would be our target space $\mathcal{Y}$, e.g. {`positive, negative, neutral`}. A model is created by a learning algorithm from a dataset D of pairs $(\mathbf{x}, y)$, typically assumed to be drawn independently and identically distributed (i.i.d) from some joint distribution $P(Y, X)$. In many learning algorithms, the model m takes the form of a conditional probability estimator $m(\hat{y}|\mathbf{x})$, which, given a movie review $\mathbf{x} \in \mathcal{X}$, would assign a probability to each sentiment $y \in \mathcal{Y}$. The goal is for m to be as close as possible to the true distribution $p(y|\mathbf{x})$, also called the target function. We use a bold $\mathbf{x}$ because the input space is usually vector valued, and we use a hatted $\hat{y}$ to emphasise that the model makes estimations that might differ from the ground truth y.

In formalising evaluation as prediction, we will simply adopt the same framework. We will define an input space, a target space, and we will see evaluation as a learning algorithm –or more generally a modelling process–, turning empirical observations into a corresponding predictive model: a score predicting model. Let's dive in, starting with the data.

3.1.1 Learning Data

Evaluation as a modelling process learns from observations of performance, which are represented by what we will call *evaluation records*. They are tuples $\langle \mathbf{m}, \mathbf{x}, r \rangle$, where $\mathbf{m}$ is the system under evaluation, which we will refer to as the *subject system* or *subject*, while $\mathbf{x}$ is an instance from the subject's input space $\mathcal{X}$, and r is some abstract score representing the quality of the subject's behaviour. In other words, an evaluation record registers how well a particular system performed for a particular instance. During a an evaluation procedure, one builds a dataset $\dot{R}$ of evaluation records, which acts as a learning base for a score predicting model. For example, when you have prompted several subjects with the instances from a hold-out test set, collected their outputs and scored them, then each score would correspond to an evaluation record, and thus a row in $\dot{R}$. Figure 3.1 shows an example.

An evaluation dataset typically contains records for multiple subjects, as we tend to compare different subjects before deploying one. Similarly, the data can – and often does– also contain records for multiple tasks, e.g. for a composite benchmark like SuperGLUE (A. Wang et al., 2019), particularly when a flexible representation for inputs is used, e.g. natural language or images.

Evaluation Records			
Subject Features (model, #params, #tokens)	**Instance Features** (what the subjects see)	Model Outputs (optional)	**Score** (a real number)
GPT-4, 1.8T, 2T	What movie do these emojis represent? 👦👴🚗⌚		0
GPT-4, 1.8T, 2T	What movie do the emojis 🔍🐠 symbolise?		1
GPT-4, 1.8T, 2T	Translate "Can I have the bill please?" into Italian.		0.7
GPT-4, 1.8T, 2T	Translate "Should I put peas in paella?" into Spanish.		0.3
Llama 2, 70B, 1.5T	What movie do these emojis represent? 👦👴🚗⌚		1
...	...		...

Figure 3.1: An example set of evaluation records. We see the scores of two large language models, GPT-4 and Llama 2, in two different tasks, one where the movie corresponding to some emojis needs to be found, and another being general translation task. In applicable scenarios, it can be worth storing the model output so different scores can still be calculated later on.

3.1.2 Input Space

In its simplest form, we can represent the problem instances with an input space that is the same for both the score predicting model and the system under evaluation. If the input space of the subject is $\mathcal{X}$, then the input space of score predicting model is also just $\mathcal{X}$. In our example of sentiment classification, the input space would be textual movie reviews for both subject and score predictor.

But not all AI systems have an input space that characterises a problem instance in way that also fits a score predictor. For example in sequential decision problems like robotics or reinforcement learning, we would have to characterise the instance with a representation that abstracts over time-steps to be able to predict performance in advance, which would be very different compared to how the subject observes the instance. It can also be hard to unambiguously define what an instance is under the presence of minor stochastic variations or complex real world conditions, e.g. what is an instance of the autonomous driving problem? For those problems, we can be pragmatic and try to delimit and represent instances with features that have an impact on performance. We will still represent it with $\mathcal{X}$, and make it clear when we need to differentiate from the subject input space.

As displayed in fig. 3.1 and mentioned before, evaluation also typically considers multiple subjects simultaneously so we can compare them, e.g. we will consider

different architectures, or training runs with a different number of epochs. We therefore consider the input space of a score predicting model to be $\mathcal{M} \times \mathcal{X}$, containing $\mathcal{M}$ representing the space over possible subjects, with elements $\mathbf{m}$. The subject feature vector $\mathbf{m}$ could be anything: a simple identifier, a comprehensive set of hyper-parameters, or even a summary representation of the training data like in Ilyas et al. (2022). We will typically consider high-level features (e.g. architecture) and hyper-parameters that allow the score predictor to differentiate between the subjects under consideration and are important for performance, often amended by a unique identifier that acts as 'catch all' to which the score predictor can attribute differences in behaviour among subjects.

3.1.3 Target Space

The goal of evaluation is to anticipate the performance of a subject, and that quality is measured by various possible scores assigned to its behaviour. A score can be anything, as long as it conceptually represents some quality. The binary correctness of a classification, cross-entropy, reinforcement learning reward, or even the toxicity of a language model output all fit the bill. This abstract notion is needed to allow our analyses to apply to a wide range of domains and use cases. We say quality of 'behaviour' rather than of 'output' to better reflect that scores can capture things like energy usage or runtime, which can differ for the same output, and that the multiple outputs produced in sequential decision making problems such as reinforcement learning or multi-step conversation are intuitively considered jointly as a 'behaviour'.

The target space for a score predictor is then equal to the range of values the chosen score can take, i.e. the negative real line $\mathbb{R}_{\leq 0}$ for a cross-entropy loss, or the discrete set $\{0, 1\}$ representing binary correctness. Unless otherwise noted, we use r to represent scores, and will use $\mathcal{R}$ to represent an abstract space of scores, which differentiates from the targets y in target space $\mathcal{Y}$ as used for subjects.

There is good reason to considering modelling scores to be non-deterministic in function of $\mathbf{m}$ and $\mathbf{x}$. For example, subjects can be configured to be stochastic, e.g. chatbots often have a parameter that controls the randomness of the output, and in general, representation $\mathbf{m}$ and $\mathbf{x}$ can fail to reflect the full complexity and detail of reality. Section 4.3.3 talks more about this. The take-away is that, regardless of wanting to represent confidence, it can be worthwhile to consider a probabilistic target space $\mathbb{P}_{\mathcal{R}}$.

3.1.4 Framework Formulation

If we now bring everything together, an evaluation protocol is a learning algorithm, learning a score predicting model $s : \mathcal{M} \times \mathcal{X} \rightarrow \mathbb{P}_{\mathcal{R}}$ from a dataset of evaluation records $\dot{R} = \{\langle \mathbf{m}_1, \mathbf{x}_1, r_1 \rangle, ...\}$, which are observations of performance r of the systems under evaluation $\mathbf{m}$, on various instances $\mathbf{x}$. If the score predictor is a probabilistic model, it is an approximation of the conditional density of the expected quality of behaviour:

$$s(\hat{r}|\mathbf{m}, \mathbf{x}). \tag{3.1}$$

This places the following question central to evaluation: given an AI system represented by $\mathbf{m}$ and an instance represented by $\mathbf{x}$, what is the expected quality of the system behaviour $p(r|\mathbf{m}, \mathbf{x})$? By observing subject behaviour for various tasks, evaluation procedures build models that answer that question.

This is a distinct instance-level perspective that allows us to closely match the formalisms of machine learning and predictive inference. There are only conceptual restrictions on input and target spaces, but there really is no technical difference between predicting scores in $\mathcal{R}$ versus predicting classification labels in $\mathcal{Y}$. This will allow us to borrow the techniques and insights from learning theory, and apply them broadly to evaluation.

3.1.5 Assessors

Score predictors exist in various breeds and families, inevitably pushing against the boundaries of our framework. For example, there is the traditional evaluation setup with a hold-out test set and aggregate metrics, which averages over and plainly disregards all instance features. It is a simple summary statistic that does not produce refined instance-level scores: predictions for all instances are equal. On the opposite side of the spectrum there is the notion of difficulty metrics, which quantify how hard an instance is, and which average over all subjects instead, lacking any refinement in the subject dimension.

We will use the term *assessor model* or *assessor*, which we introduced in Hernández-Orallo et al. (2022), to refer the class of score predicting methods that are a direct translation of our framework. They produce instance-level scores for multiple subjects, with refinement across both dimensions. Figure 3.2 provides a diagrammatic overview.

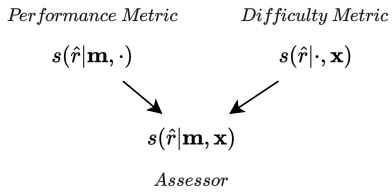


Figure 3.2: Assessor models as the mix between subject focused and instance focused score predictors. The central dot '$\cdot$' represents aggregation over the respective argument.

Assessors represent the very general method of taking a dataset of evaluation records $\dot{R}$, accepting $(\mathbf{m}, \mathbf{x})$ as an input and predicting any score r as a target, and training through supervised machine learning algorithm on evaluation records $\dot{R}$. This exact method has been used in practice and in different unconnected research efforts, for example in Shnitzer et al. (2023) or Ong et al. (2024) for application to instance-level model selection (section 4.1.3.2). Section 4.2.5.3 will talk more about assessors.

3.1.6 An Example

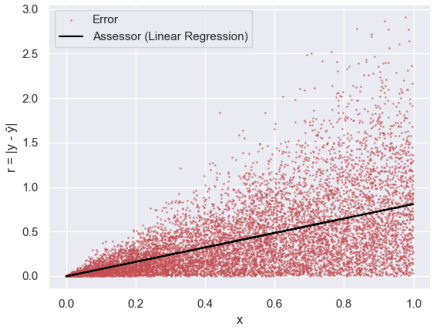
Because all of this has been terribly abstract, it is urgently time that we show an example. Figure 3.3 shows a synthetic regression scenario, where $y = 2x + \epsilon$ with $\epsilon \sim N(0, x)$. The fact that the noise ϵ depends on x means that it is heteroscedastic, i.e. with unequal variance. As a result, the linear regression performs better in some regions than others: it has higher error for higher x. Panel (b) shows the distribution of the subject's errors, and a linear regression score predictor fitted to it.

The presence of a pattern in the error distribution allows us to meaningfully use the score predictor. For example, to provide a naive prediction interval, as shown in panel (c), or to do performance estimation under distribution shift, as shown in panel (d). In reality distribution shift will not be that extreme, but the point stands: by providing instance-level performance estimation, we can provide an accurate assessment of the performance under co-variate shift. The Mean Absolute Error (MAE) based on assessor predictions is 0.65, closely matching the actual MAE of 0.64 during deployment, while the MAE of 0.40 calculated during evaluation suffers in precision due to the distribution shift.

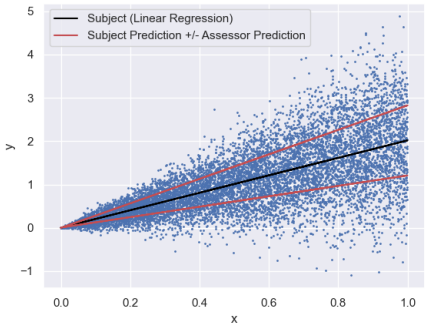
This functionality could also have been provided by the subject if it used a probabilistic model $p(y|x)$ rather than the linear regression used now. But probabilities

are not always an option or might require more complex statistical methods, while in our demonstration case the score predictor is in the same simple class of linear regressors.

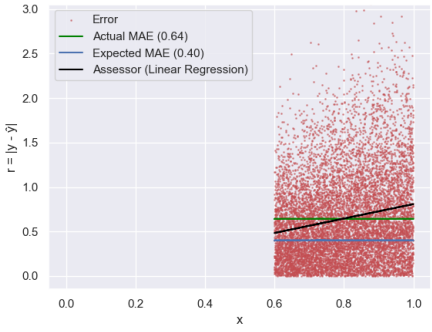
(a) Example subject regression



(b) Example score predicting regression



(c) Naive prediction interval



(d) Distribution shift

Figure 3.3: Example Score Prediction Application. Panel (d) illustrates a co-variate shift in distribution. The assessor model has been fitted based on panel (b), as has the 'Expected MAE' represented by the blue line. Inspired by and adapted from NannyML (2024).

3.2 Evaluating Evaluation

Between the different breeds of score predicting methods, some will work better than others. But how do we measure and compare them? In other words, how do we evaluate evaluation? Before talking about the various sorts of score predictors (sections 4.3 to 4.2), we will first talk about how to measure their score predicting capabilities.

3.2.1 Evaluation Error

In a supervised prediction problem –like the sentiment classification example of the previous section–, the generalisation error is a measure of how accurately a subject is able to predict outcome values for previously unseen data. It is typically formulated as the expected value of a divergence $\mathcal{D} : \mathbb{P}_{\mathcal{Y}} \times \mathbb{P}_{\mathcal{Y}} \to \mathbb{R}_{\geq 0}$ over some assumed distributions X and Y. With $\mathbf{m}$ being the AI system under evaluation and p being the true distribution, it looks like $\mathbb{E}_{X,Y}[\mathcal{D}(m(\hat{y}|\mathbf{x}), p(y|x))]$.

Since we realistically work with a finite amount of data, we have to approximate, and we measure the empirical error instead. This is the average of a loss $\mathcal{L} : \mathbb{P}_{\mathcal{Y}} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$ over a finite amount of samples D: $\frac{1}{|D|} \sum_{\langle \mathbf{x}, y \rangle \in D} \mathcal{L}(m(\cdot|\mathbf{x}), y)$, with D for example being a hold-out test set.

For score predictors the principle is exactly the same. We would like to calculate the evaluation risk,

$$\epsilon_{X,M}[s] = \mathbb{E}_{X,M}[\mathcal{D}(s(\cdot|\mathbf{m}, \mathbf{x}), p(r|\mathbf{m}, \mathbf{x})], \tag{3.2}$$

but instead, using a hold-out set of evaluation records $\dot{R}$, we must calculate the empirical evaluation error:

$$\epsilon_{\dot{R}}[s] = \frac{1}{|\dot{R}|} \sum_{\langle \mathbf{m}, \mathbf{x}, r \rangle \in \dot{R}} \mathcal{L}(s(\cdot|\mathbf{m}, \mathbf{x}), r). \tag{3.3}$$

We will use the term *evaluation error* to refer to the prediction error of a score predicting model, and analogous to the subject's prediction problem, we could also calculate various other relevant metrics, i.e. evaluation accuracy or evaluation Area Under the Receiving Operating Characteristic curve (AUROC).

Let us go back to our sentiment classification example, and imagine we have calculated the accuracy of the subject to be 83%, meaning that, for 83% of instances,

our subject classifies the sentiment of the review correctly, and receives a corresponding score of 1. This metric now acts as a score predictor, telling us what to expect when we try the subject with a new instance. It tells us that for any instance, the chance of success equals 83%. As a 'model', a summary statistic is fantastically simple, interpretable, and often the only produced artefact of an evaluation, as it often is sufficient for making the decision of whether to deploy the model or not. But it should be clear that as a score predicting model it leaves much on the table in terms of prediction error. While the statistic is perfectly calibrated, meaning the predicted chance of success matches the actual chance of success, there is no form of refinement or discrimination between instances. It would be like a weather forecast simple reporting what the average precipitation is for this time of year, not taking into account current atmospheric conditions at all. If we hope to better understand where and why our subjects do well, we must raise the bar.

3.2.2 Higher-Order Evaluation

As you might have noticed, we have found ourselves in a little recursion. If we consider evaluation as score prediction, and we need to evaluate score predictors, that evaluation is again a score predictor, continuing ad infinitum (and ad absurdum). Table 3.1 provides an overview of this nesting of nth-order predictive models. Luckily, so far we have found no literature that went multiple levels of 'meta', with 2nd-order evaluation being the highest with analysis of quantitative nature in practice. The higher up, the simpler the models tend to become, and when the score predictor is sufficiently simple, e.g. a summary statistic, instead of using empirically collected data points we rely on theory and assumptions to evaluate it. In other words, at a certain point, human mental models take over from explicitly calculated statistical models: we do not use a hold out set to test the predictive power of a summary statistic, we assume that we understand how the statistic behaves.

3.2.3 Data Requirements

Higher orders of evaluation need progressively larger orders of magnitude of data to be significant. If we consider a testing ratio of 1/5th, i.e. a standard 80/20 train-test split, with a total of 25 000 available instances, then 5000 would be used for testing the subject, of which only 1000 would be used for the 2nd-order evaluation. For a test ratio of $1/t$, an n-th order evaluation, and a desired test set size of c, a total number of $c \cdot (1/t)^{-n} = c \cdot t^n$ instances are needed.

Evaluation Order	Predictive Order	Examples
/	1st-order predictive model	subject model solving the original task
1st-order evaluation	2nd-order predictive model	aggregate metrics, assessors, scaling laws
2nd-order evaluation	3rd-order predictive model	metrics on the assessor or scaling law
3rd-order evaluation	4th-order predictive model	mental models of higher-order evaluation
...	...	...

Table 3.1: Different orders of evaluation with example procedures. We can conceive of realistic examples of higher order predictive models until at least the 4th order. Scaling laws are discussed in section 4.2.3.1 and assessors in section 4.2.5.3. A 1st-order evaluation produces a corresponding 1st-order score predictor, which is a 2nd-order predictive model.

There is another consideration as well: data imbalance. The data balance for a score predictor is determined by the performance of its subject, and subjects that perform badly are usually not of interest. A subject with 90% accuracy on a classification problem will entail 10/90 imbalance ratio for the score predictor. Failure cases are usually also more informative. If we would want to see c_e cases with bad performance, and this happens only in $1/e$ cases (across orders), we need $c_e e t^n$ instances, i.e.,

$$c_e = \frac{|D|}{e t^n}. \tag{3.4}$$

So for the same 25 000 total instances, at 90% accuracy at all levels ($e = 10$), and 80/20 test split ($t = 5$), we would see only 100 bad predictions during 2nd-order evaluation, i.e. when evaluating a 1st-order score predictor. This can be problematic for evaluating score predictors with data intensive evaluations, such as the measurement of calibration at high confidence.

3.3 Prediction and Predictability

If we want to determine how good a certain score predictor is, i.e. we want to do 2nd-order evaluation, then we calculate the empirical error from eq. (3.3) over a dataset $\dot{R}$. Now, that results depends on the subjects producing the scores contained in $\dot{R}$: some subjects might be more predictable than others, e.g. a subject that is always wrong, is very easy to predict scores for. This is why, apart from a distribution X over instances, we also consider a distribution M over subjects in eq. (3.3).

But as opposed the task represented by X, which is usually given and fixed, in many applications we have control over the subject: we can choose which ones to deploy and which to discard. And we care about the predictability of the subject performance as a useful property in its own right: the more predictable a subject is, the more we can control its behaviour, for example by avoiding to use it for a certain when the corresponding expected score is below a certain threshold.

3.3.1 Subject Predictability

But how do we measure predictability? Much like how the difficulty of a task is measured with respect to a specific class of models, see e.g. Ethayarajh et al. (2022), to define the predictability of a subject, we must consider a class of score predicting models first. For a given subject m represented by $\mathbf{m}$, predictability with regards to family $\mathcal{S}$ of score predictors and a data distribution X, is then defined as:

$$P_{\mathcal{S},\mathcal{X}}(m) := \max_{s \in \mathcal{S}} \mathbb{E}_{\mathcal{X}}[1 + \mathcal{D}(s(\hat{r}|\mathbf{m}, \mathbf{x}), p_{\mathbf{m}}(r|\mathbf{x}))]^{-1}, \tag{3.5}$$

with $\mathcal{D}$ being an appropriate divergence, with range $[0, \infty]$ and higher numbers indicating more divergence. This is formula reflects finding the lowest possible evaluation error, and to make predictability positively-oriented (meaning higher being better) in the range $[0, 1]$, we take the inverse, and therefore also the maximum.

The family $\mathcal{S}$ could be a specific parametrised class of functions, a family of learning algorithms, or even be defined based on a maximum computational or data budget, e.g. score predictors of similar scale as the subject model. The value is uncomputable given the expectation over $\mathcal{X}$ and the practical infeasibility of iterating over realistic versions of $\mathcal{S}$, but we can calculate a reasonable approximation by considering a fixed population of score predictors. We take a maximum, rather than an average, as the average is likely to be dominated by non-interesting or badly performing score predictors in the family $\mathcal{S}$, and for practical purposes, it is sufficient to define the predictability of a subject as the performance of the best score predictor that can be found. Ethayarajh et al. (2022) have a similar definition for dataset difficulty, and fig. 3.4 displays the general symmetry between the different levels of prediction.

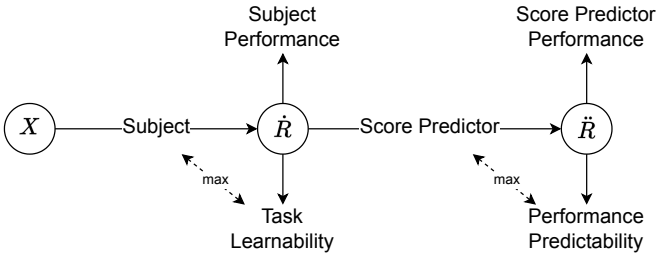


Figure 3.4: Co-dependencies of predictability and score predicting power. We can not measure predictability of evaluation results $\dot{R}$ directly, and instead must measure it through a score predictor, the same way the learnability of a task X must be measured through a subject. Circular nodes represent random variables, while their inter-dependencies are represented by solid arrows, each corresponding to a functional mechanism transforming the variable. Based on the random variables, four metrics can be defined, with the dashed connectors indicating that to calculate the respective metrics, one must search over corresponding mechanism that would maximise it.

3.3.2 Balancing Predictability and Performance

Note that the predictability value of eq. (3.5) can be maximised by degenerate cases, such as subjects with zero performance. For a subject to be useful, one must strike a balance between a subject's performance and its predictability, and improving one might damage the other.

There is a rich area of literature to borrow from here: machine learning with a reject option (Hendrickx et al., 2024). As discussed in section 2.4.5, a 'rejection' is the choice to abstain from using a subject prediction because the expected chance of it being correct is too low. We can however not adopt their measures directly. This for two reasons: (i) reject options are dominantly studied for subjects that are classifiers, while considering only binary scores, i.e. assign 1 if the classification is correct, and 0 if not; and (ii) many metrics assume that both subject and score predictor are fixed and given, or that the rejection rate is fixed and given.

The latter, exemplified by the metrics in Condessa et al. (2017), is useful when we only want to report the quality of specific pair (m, s), but is not fit for comparing between pairs of (m, s), i.e. when either $\mathbf{m}$ or s is free to pick, which is the case for many applications. Take for example 'Classification Quality' from Condessa et al. (2017), adopted as 'Combined Quality' in Hendrickx et al. (2024). Seeing it as a function of a dataset D, subject m, and score predictor s, it can be formulated as such:

$$CQ(D; m, s) = \frac{\sum_m \mathbb{1}[s(m, x) = \dot{g}(m, x)]}{|D|}, \tag{3.6}$$

where $\mathbb{1}$ is the indicator function, $\dot{g}$ is a scoring function, and both s and $\dot{g}$ have a binary range. But it is actually a measure of the performance of the score predictor, and like eq. (3.5), this equation is also maximised by a subject with zero performance, which is perfectly predictable. All metrics in Condessa et al. (2017) and Hendrickx et al. (2024, Sec 3.2) are maximised by degenerate cases: either maximally conservative score predictors, maximally optimistic ones, or zero-performance subjects.

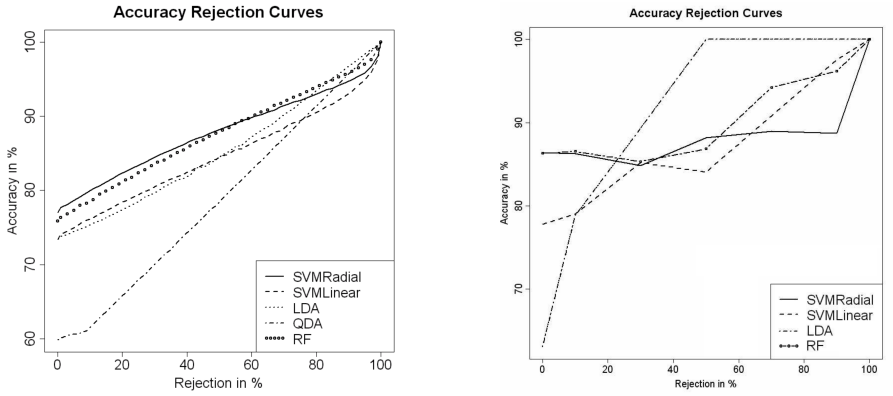


Figure 3.5: An example Accuracy-Rejection Curve. Both panels show a classification setting with a reject option, comparing different `(subject, score predictor)` pairs. From Nadeem et al. (2009), published in PMLR under CC BY-SA 4.0, no changes made.

A better option is the Accuracy-Reject Curve (ARC), as proposed in (Nadeem et al., 2009), which are closely related to the Error-Reject Curves (ERC) in Hansen et al. (1997). An ARC plots the performance of the subject against the rejection rate based on score predictor confidence. Figure 3.5 shows an example ARC plot. We can take the area under the ARC to have a measure of (m, s) that reflects both subject performance and its predictability. The ARC can be easily extended to any problem with binary correctness, not just classification. But we can also extend it to apply to any positively oriented score, where the y-axis would reflect the average score, and the rejection rate on the x-axis is determined by the threshold on the predicted score. The area under ARC (AUARC) does not measure calibration of the score predictor, only refinement.

For high-risk applications, we are particularly interested in the length of the x-axis segment where the ARC touches the top of the plot. This is commensurate with what we will call the 'Perfect Predictability Region' (PPR): the area of the input space where the score predictor has no false positives, i.e. the score predictor predicts success and the subject is always correct. Assuming binary scores, the equation looks like this:

$$\begin{aligned} PPR(D; m, s) &= \frac{1}{|D|} \max_{\tau}(\pi \cdot \sigma) \\ \text{with } \pi &= \prod_{x \in D} \mathbb{1}[s(1|m, x) \geq \tau \Rightarrow \dot{g}(m, x) = 1] \\ \text{and } \sigma &= \sum_{x \in D} \mathbb{1}[s(1|m, x) \geq \tau \wedge \dot{g}(m, x) = 1]. \end{aligned} \tag{3.7}$$

Note that if at any point s predicts success while m is actually wrong, then π will be zero. A single high-confidence misprediction by the score predictor can thus seriously impact the PPR. The maximisation will select the lowest τ for which $\forall x\ \dot{g}(m, x) = 1$ holds. If we measure a PPR of 0.2 or 20%, we know that we can deploy the subject, get predictions for 20% of the instances, and make zero mistakes. The formula can also be adapted to be more lax, e.g. tolerating up to 1% errors, by modifying π appropriately. The example to the left in fig. 3.5 has a PPR of near zero, while the example to the right has a PPR of approximately 50%. In general, PPR is expected to be low, and with high variance if it would be high.

3.4 Interlude: Expanding Score Prediction with Users

In Schellaert et al. (2023) we argue that modern GPAI systems should be considered to be cognitive extenders –tools that become a literal part of an human's mind– much like as has been argued for simpler tools such pen and paper or calculators (A. Clark & Chalmers, 1998; Menary, 2010).

The utility of a such a cognitive tool m strongly depends on the user h interacting with it. Different users want to complete different tasks, and tool performance can differ over these tasks. But even if the tasks are equal, users have different ways of interacting with the system, and the interaction style impacts performance. Like how a hand saw will cut more efficiently with long strokes, so also do GPAI systems work better with particular usage, as is evident by e.g. prompt engineering practices (Sahoo et al., 2024), and facilitated by the flexible interface that is natural language.

A user needs to map its abstract concept x of the problem at hand to a concrete representation $w(x) = \mathbf{x}$ that can serve as an input to m, where w represents that wrapping process. Similarly it must interpret the subject's behaviour b through an unwrapping process $u(b) = y$. Figure 3.6 illustrates all the steps involved with system interaction.

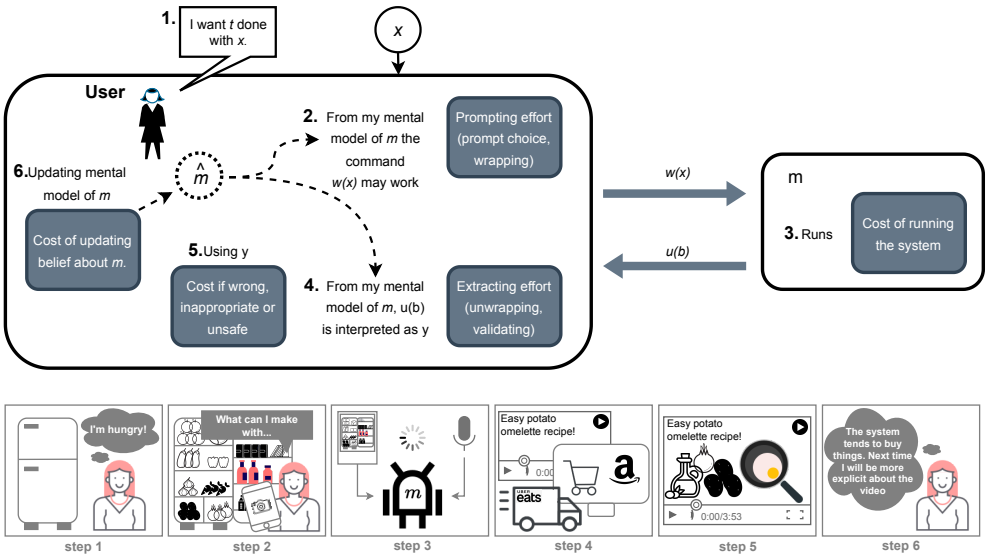


Figure 3.6: Elements of the prompting process. Top: The different steps of human user directly prompting a subject m. The user wants a task t done (1) possibly including some multimodal elements x and, according to the internal model of m (denoted by $\hat{m}$) has to articulate the wrapping of these elements $p = w(x)$ into a prompt for the system (2), producing an output b (3) from which the user unwraps the result $y = u(b)$, validates/assesses it (4) before finally using it (5). Given this iteration, the user updates (6) the internal model of m. Bottom: Figurative example of a subject m generating a recipe video from a photo of the contents of a user's refrigerator and a voice prompt. The user needs to deal with several outputs given by m, including not only a recipe video generated by m but also some other continuations such as the suggesting of ordering of more ingredients or ordering the whole dish to a restaurant, which she had to stop. After this interaction, the user updates her model of the system ($\hat{m}$) thinking she should be more explicit about wanting a recipe video next time.

The relevant implication for score prediction is that, even if a GPAI systems deterministically maps from inputs $\mathbf{x}$ to behaviours y, a particular user might have more trouble coming up with an adequate representation $\mathbf{x}$, and as a consequence, *assign a different score*. The same goes for unwrapping behaviour and other

steps in the prompting process: successful interaction has different prevalence for different users, and thus different reward. This causes variance in the distribution $p(r|\mathbf{m}, \mathbf{x})$, which is what we are trying to model during score prediction. *As a consequence, to do precise score prediction, we need to model* $p(r|\mathbf{m}, \mathbf{x}, \boldsymbol{h})$ *instead.* In fig. 3.7 we show an updated version of fig. 3.7.

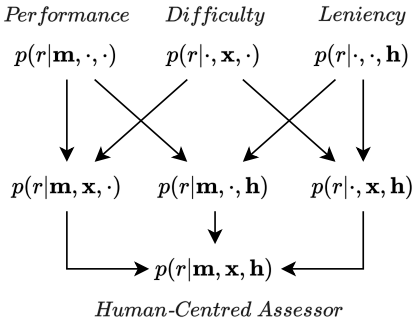


Figure 3.7: Human-Centred Assessors as the mix between performance-, difficulty- and leniency metrics. The central dot '·' represents aggregation over the respective argument.

For the rest of dissertation, we will abstain from including **h** explicitly to lower notational overhead, as it should not impact further theory.

3.5 Revisiting Evaluation Challenges

Figure 3.8 presents an overview of evaluation as a causal graph, summarising our proposed framework. The goal is to predict R', since we assume we cannot calculate $\dot{g}(X', Y')$, as that requires observing subject behaviour $y' = m(x')$, which might be dangerous or otherwise costly. The application section (see section 4.1) will motivate this assumption –or rather this constraint– further.

> **What about Y?** Note that we have left out label distribution Y from the graph. This is because we also aim to cover non-supervised problems without a target label, e.g. sequential decision problems like robot control, or text2image generation. A separate label variable can easily be added in: we can imagine an extra Y' and Y together with X' and X, which are connected to the same parent and children nodes, except they are not connected to the predictions $\hat{Y}'$ and $\hat{Y}$, nor to $\hat{R}'$.

To start dissecting the diagram, we can analyse how the default evaluation procedure, i.e. the aggregate metric with a hold out test set, fits in. Ignoring the intermediary evaluation records $\dot{R}$, which overcomplicate things in this particular scenario, the predicted score $\hat{R}'$ looks like this:

$$\hat{R}'_{agg} = s_{agg}(\mathbf{m}, X') = \mathbb{E}[\dot{g}(X_E, m(X_E))]$$

As we see, and not surprisingly, X' is entirely ignored. The aggregate metric is not conditional on the deployment data, and relies on $X_E \sim X'$, i.e. both variables being being i.i.d. Now, in a way this is obvious. A metric is not a model. But we need a model! We need generalisation and some way to condition on X'. It is not realistic to assume $X_E \sim X'$, especially as we start to deploy AI systems in the open world, expecting users to extrapolate reported performance to their specific scenarios.

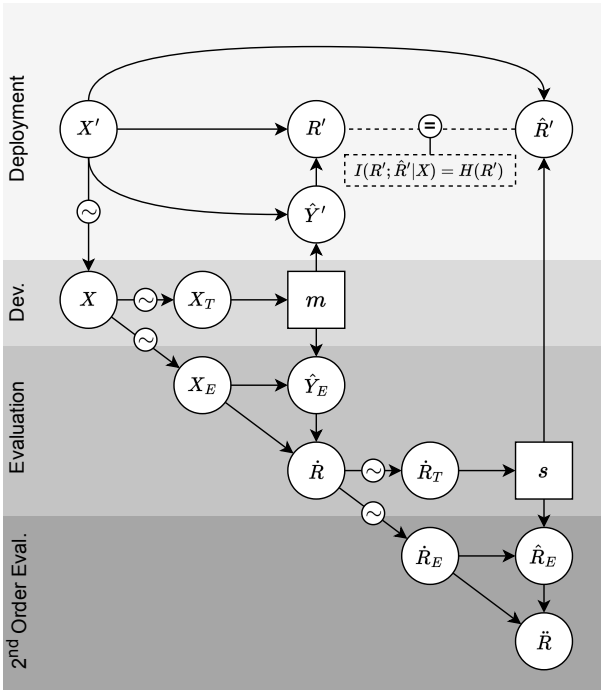
Figure 3.8 also encodes the commonly made assumptions required to make evaluation be accurate. The diagram allows us to pinpoint exactly which assumptions break for various problem in evaluation. Using this overview, let's revisit the challenges mentioned at the start of this dissertation –those in sections 2.5 and 2.6 about refinement, distribution shift, contamination etc.– to view them in a new light.

Some of them can be remedied by conditioning on X', others not. We will start with the former.

3.5.1 Distribution Shift

Let's start with distribution shift in the input variable X', also called co-variate shift in supervised learning. Co-variate shift means the i.i.d. assumption $X, X_T, X_E \sim X'$ does not hold, because the variables are not identically distributed, i.e. $P(X) \neq P(X')$. There is some other data generating process, which is not of interest, from which X is sampled.

But we can expect some overlap: the mutual information $I(X'; X)$ is rarely zero, and therefore neither is the mutual information $I(R'; \dot{R})$. That means the score predictor s can still learn useful patterns. Especially if the possible outcomes of X', i.e. $\{x' : P(X' = x') > 0\}$, are well covered by the test set, then the distribution shift does not entail out-of-distribution realisations and the score predictor can learn everything that is needed by modelling $P(\dot{R}|X_E)$.



$$
\begin{aligned}
X' &= U' \\
R' &= \dot{g}(X', \hat{Y}') \\
\hat{R}' &= s(\mathbf{m}, X') \\
\hat{Y}' &= m(X')
\end{aligned}
$$

$$
\begin{aligned}
X, X_T &= U' \\
m &= \text{train}(X_T) \\
\hline
X_E &= U' \\
\hat{Y}_E &= m(X_E) \\
\dot{R}, \dot{R}_T &= \langle X_E, \dot{g}(X_E, \hat{Y}_E) \rangle \\
s &= \text{train}(\dot{R}_T) \\
\hline
\dot{R}_E &= \langle X_E, \dot{g}(X_E, \hat{Y}_E) \rangle \\
\hat{R}_E &= s(\mathbf{m}, \dot{R}_{E,X}) \\
\ddot{R} &= \ddot{g}(\mathbf{m}, \hat{R}_E)
\end{aligned}
$$

Figure 3.8: Causal Graph of Evaluation. Circles represent random variables, while squares –notably subject model m, and score predictor s– represent probabilistic models or estimates, which as a product of a learning algorithm based on random variables, are also random in nature. The random variable X represents instances, while X_T and X_E are train and evaluation instances respectively. Further down the line, $\dot{R}$ represents evaluation records, and $\dot{R}_T$ and $\dot{R}_E$ are again train and evaluation instances, this time for the score predictor. $\hat{Y}$ represents subject outputs, and $\hat{R}$ score predictor outputs. All variables with an apostrophe, e.g. Y' represent the same random variables as originally, but now 'during deployment', i.e. when we are relying on the model's inference to make decisions, and where the distributions over the variable might have changed. The arrows represent causal influence, with the similarity signs ($\sim$) representing the assumption that the variables independent and identically distributed (i.i.d). At the top right of the diagram, the equal signs represents the goal of of achieving high mutual information between predicted scores and actual scores, conditional on X'. The left side is divided in several stages of the AI pipeline: development (or training), evaluation, and deployment. For completeness, although not often considered explicitly, we present second order evaluation as well. The right side of the final reflects how the variables are functions of each other, with X' coming from an unknown data generating process represented by U'.

Condition on X' is needed though, as while $P(\dot{R}|X)$ might resemble $P(R'|X')$, we should not expect the same for the unconditioned $P(R')$. This is what we demonstrated in fig. 3.3, where the score prediction error under distribution shift was very low for the assessor model, but very high for the aggregate metric.

We will consider both label shift and anti-causal notions of distribution shift, i.e. concept shift and manifestation shift, as described in Murphy (2023, Sec. 19), which are largely specific to supervised learning, to be out of scope for now.

3.5.2 Refinement

Refinement (section 2.5.3 is the best demonstration of why conditioning on X' is important for a score predictor. Remember, with refinement we refer to notion that we want to predict the score for a particular question, i.e. a particular realisation x' of X'. In other words, we care about $\mathbb{E}[R'|X' = x']$, not just $\mathbb{E}[R'|X']$.

If we do not condition the score predictor on X', we cannot take into account x to to estimate $\mathbb{E}[R'|X' = x']$.

> **Note:** Refined prediction can also be considered an extreme form of distribution shift, where $X' = \delta(x)$, with δ referring to the Dirac delta function.

3.5.3 Lack of Prediction Target & Increase in Scope

The lack of prediction target, both for evaluating learning algorithms (section 2.5.4) and evaluating GPAI systems (section 2.6.1), refers to the idea that there is an unknown downstream use case X', which, because it unknown, entails $X \not\sim X'$.

To explain further, let us first consider X' as representing the (unknown) hierarchical distribution over all downstream use case X'_1, X'_2 and so on. There are two related problems. One is estimating which subject is the best overall, i.e. considering X' in full. The second is estimating performance for a specific use case X'_i.

For any realistic input space $\mathcal{X}$, the entropy $H(X')$ is huge, as for a general-purpose tool, the set of possible realisations $\text{supp}(X') = \{x' : P(X' = x') > 0\}$ is huge. The scope of the evaluation is then correspondingly large: the test set X_E (and corresponding evaluation records $\dot{R}$) needs a support $\text{supp}(X_E)$ that

is somewhat proportional in size to $\text{supp}(X')$[1] if we want R' to be somewhat accurate.

For estimating $R'_{(i)}$, the problem is however exactly the same. Even if we have a score predictor that can condition on X, for the score predictor to provide accurate predictions for an arbitrary case $X'_{(i)}$, the demands on evaluation are equally high. In any realistic case, we need a big test set, plus for to score predictor to have some form of generalisation.

This is why when evaluating these these general-purpose tools, the practice is to evaluate on a collection of datasets and benchmarks, preferably as big as possible. Afterwards, it is usually humans doing both the conditioning and the generalisation.

And this is where the difference between evaluation learning algorithms and GPAI systems starts to show, and the problem worsens. When we evaluate learning algorithms, we do not need to be particularly accurate, because as long as $\hat{R}'$ correlates to some regard with R' over different algorithms m, the evaluation serves its purpose: the users doing the conditioning and generalisation are AI developers that are experts *and* run additional evaluation for their specific use case $X'_{(i)}$ as they still need to train a concrete m. A rough score estimate is sufficient, so they can decide whether the use-case is feasible and select worthwhile learning algorithms to try out.

But for GPAI systems, that is not the case! Non-experts use these systems directly, without any prior expertise in AI, with anecdotal evaluation from a few interactions. They have trouble interpreting benchmark scores, not knowing how the results generalise to their use case. They do not know how different AI systems relate to each other, and what their inherent weaknesses are. Even experts have trouble consolidating and interpreting all of the evaluation information involved with a modern LLM benchmark such as HELM (Liang et al., 2022).

This also explains the success of 'in-the-wild' collections of user queries and corresponding scores through efforts such as ChatBotArena (Chiang et al., 2024). Although not refined to specific $X'(i)$, these are a more natural representation of X' than other benchmarks in the literature.

Better epistemic or tooling support from evaluation science is needed: e.g. machine score predictors that generalise, with flexible input spaces to describe the use case $X'(i)$ with something else than samples; or score predictors that can

[1] Actually it is the $|\text{supp}(X_E) \cap \text{supp}(X')|$ that needs to be proportional, but we expect that to equal $|\text{supp}(X_E)|$ due to the all-encompassing nature of $\text{supp}(X')$

provide refined predictions given realisations x; or maybe just more powerful and interpretable testing procedures, that allow humans –expert or not– to generalise more confidently.

Note: Developers of GPAI systems use a sizeable sample from X_T: large collections of internet text, multiple hundreds of Gigabytes in storage size. They also have a pretty good idea of X', as they can –barring data-protection constraints– observe the queries x their users send them. But what they do not have is a proportionally sized sample from X_E, as they do not necessarily observe the scores users give to the system outputs, although many have started incorporating feedback mechanisms. We learn here that to create predictable AI systems, it is recommended to balance the size of training and test set.

3.5.4 Test Adaptation & Contamination

There are also problems in evaluation that have less to do with conditioning on X.

Recall, test adaptation occurs when subject developers get repeated exposure to the test set. This entails a new connection $\dot{R} \rightarrow m$, causing $m = \text{train}(X_T, \dot{R})$. Contamination refers to the leakage of items from the test set leak into the training set, and entails a new connection $X_E \rightarrow X_T$. Both of these mechanisms are not present in deployment, resulting in $P(R'|X')$ and $P(R|X)$ to be different distributions, affecting the score predictor even if $X = X'$, i.e. without co-variate shift.

Both test adaptation and contamination are only a problem because the test set is not perfectly representative of deployment. If the deployment distribution was a uniform distribution over the test set, or even if the degree of contamination would match the between testing and deployment, there would be no issue. A score for a trained-on instance is less representative than a score for an unseen instance, as the latter better reflects the plausible scores for similar instances.

3.6 Discussion

Before we start any discussion, in case there are any questions regarding the design decisions of the framework, appendix B provides additional motivation, specifically addressing the following questions: *'Why not predict subject behaviour instead of scores?', 'Why not observe subject behaviour when predicting scores?', 'Why focus instance-level score predictions?', and 'Why would (fine-grained) score prediction even be possible?'.*

This chapter has presented a framework for conceptualising evaluation as a prediction problem, with the goal of anticipating the performance of AI systems in their deployment setting. Our particular framing centres on conditioning the prediction of scores r on a representation of the problem $\mathbf{x}$, and in case multiple subjects need to considered, on a subject representation $\mathbf{m}$.

Upsides

This conditioning on $\mathbf{x}$ essentially solves the refinement problem, barring any other challenges such as distribution shift.

But we also made headway with regard to those other challenges. Certain forms of distribution shift can be perfectly handled by score prediction conditioned on $\mathbf{x}$, although accurate prediction under others forms of shift, e.g. out-of-distribution samples, remain dependent on the precise generalisation capabilities of the score predictor. However, our close connection to machine learning and statistical inference formalisms should allow us to adopt methods from the literature, e.g. meta-learning (Hospedales et al., 2020; Vilalta & Drissi, 2002) or invariant risk minimisation (Arjovsky et al., 2020), to better deal with those more difficult distribution shifts.

And while we can not directly attack the lack of prediction target in the evaluation of learning algorithms or GPAI systems, we can now clearly see why the problem is worse for GPAI systems, where downstream testing is not the norm, and non-expert users need to make predictions about performance. We also know that solving it requires dealing with the significant increase of scope of evaluation first, for which the parallels with machine learning again show a plausible way forward: scale up the evaluation data, and create score predictors that generalise well. Following up, our definition of the evaluation error and second-order evaluation then allows us to reason about what it means for a score predictor to generalise well, and to compare different techniques.

As a result of the discussion of second-order evaluation, we also realised the difficulty of the score prediction problem significantly depends on the subject itself and the amount of pattern that is present in the relation between input and score. This identifies another road for remediating evaluation challenges: make the subjects more predictable in the first place, with a clearly defined and easily learnable area where they consistently operate well.

A proxy for contamination. While we cannot tackle contamination directly unless we can break or analyse the relation $X_E \to X_T$, we can provide for an alternate perspective: contaminated instances are those instances where a good score predictor expects low performance, but the subject nonetheless performs well. The standard idea of contamination based on instances being duplicated has two issues: (i) scores could be good regardless of contamination, which does not give the subject chance to demonstrate generalisation, (ii) one needs to look at instance similarities, not only strict equality, which worsens the previous problem.

We cannot rely on a subject's confidence here, as the subject will not be surprised by the contaminated instance –given it has seen it during training– and will thus predict with high confidence. Score prediction needs to be anticipative here! We could rely on the self-assessment of a very similar subject –but trained on different data– as a proxy, at which point that subject would act as a score predictor. Training on different data splits could also be a way to reduce or eliminate the effect of the contamination on the score predictor.

Downsides

As identified in section 3.2.3, there is a significant extra data burden to train good generalising score predictors, and especially to compare different ones during second-order evaluation. Combined with the significant increase in scope for the evaluation of GPAI systems, the efforts invested in evaluation will need to evolve into being proportional to that of training.

A major issue is that if we want the score predictor to condition on X' in the case of distribution shift, we do need some information about X'. During instance-level score prediction, this is not an issue, as we can access the realisation $\mathbf{x}$. But if we want to predict an aggregate score pre-deployment, we need to anticipate X' in some way. Sometimes we can draw unlabelled samples from X', which would allow us to use the score predictor as is. But more often than not, what we know about X' is high-level: a new hospital to deploy the cancer screening tool in, a

new city to deploy a fleet of robotaxis too... Maybe the problem is not predicting scores, maybe it is predicting the future. If it could predict the future, we could run it through our models and grade the corresponding behaviours –or at least subsample or oversample the most similar test data–, without needing to predict scores at all; a measurement will always be more accurate than a prediction.

Barring access to a magic crystal ball, solving aggregate score prediction will require building score predictors that can work with high-level descriptions of X'. In Section 4.2.5.4, we will talk about capability-oriented evaluation, which is such a method that is since recently being explored. It requires describing the task in function of its cognitive demands, e.g. the level of reasoning, memory, or navigation skills required, and if the distribution of those demands is known for the deployment distribution, we can infer scores.

Chapter 4

Applications and Methods

4.1 Applicability

The philosophers of science have made prediction a central goal of science for at least two broad-scoped reasons. One is that prediction strengthens theory. Recall from the introduction that Whewell (1849/1969) and Duhem et al. (1982) argued that scientific theories are proved, or at least reinforced, by their prediction of surprising results, and that comparing predictive ability is a valid option when choosing between competing theories. The other, more plastic and mundane, motivation is that, to paraphrase Longino (2002), prediction facilitates "intervention, control, or other forms of action on and among the objects in nature". The degree of control we have over (im)material outcomes and directions defines the degree to which we can tweak the world and the future to our liking.

The reasons for predicting performance, rather than measuring or observing it, lie in those same motivations. On the one hand there is prediction for prediction's sake: to test and expand knowledge, to guide the proposal and acceptance of theories, to reason in the counterfactual. In other words, we predict performance explicitly because we want to learn about what influences it: what makes instances hard, and subjects capable. On the other hand, there is control; because as is often forgotten in the abstract world of theory, measurement requires observing behaviour, and all behaviour has a certain effect on the state of the world. If the

effects are negligible and unimportant, we can turn to observing behaviour and measuring its quality, which is always preferable in terms of accuracy. If, on the other hand, the effects are consequential and need to be optimised, we have to predict instead.

In terms of effects, at the very least, compute resources have been spent and time has elapsed. This is in itself cause for optimisation! It can be cheaper to predict rather than to measure performance, see e.g scaling analysis (section 4.1.1.2), especially when collecting subject-instance behaviour is expensive, such as for subjects that require significant compute resources or that are behind a paid API. More prominently, we might want to select between subjects (or reject) because the decision to deploy them is a coarse one: the subject is used for several (yet unknown) instances, or not at all. We cannot observe behaviour, so we must anticipate it. This reasoning also applies to sequential decision problems consisting of multiple steps and actions. An individual action can be observed, and might cause a certain effect, but the behaviour can only be graded at the end of the task and in its entirety. In the meantime damage –read: non-optimal state change that cannot be trivially reversed– might have been done.

We have chosen to organise the applications by their broad motivation, starting with control, split in optimising development (section 4.1.1), optimising evaluation (section 4.1.2), and optimising inference (section 4.1.3), and ending with the act of knowledge building, or prediction for prediction's sake (section 4.1.4).

> **Predicting versus measuring.** We pay a lot of attention to the notion of anticipating performance and its difference with measurement and grading. In reality, the line between being anticipative or not can be blurry, as sometimes there is just no cost to observing outputs, but also no benefit. To prove the utility and validity of our research thesis, i.e. that the goal of evaluation is prediction, we must be careful not to attribute virtue where it is not deserved, i.e. where prediction is an inconsequential design decision.

4.1.1 Optimising Development

Optimising development refers to the intention of maximising (aggregate) subject performance while controlling for development time and resource costs. We are not interested in inference performance here, the metric of interest is the performance on the validation set. While performance on the validation set can be measured, we want to select and search through designs first, considering subject features such as hyper-parameters, architectures, and training data, before committing resources that might be wasted. We want to avoid training subjects that

will be discarded anyway. In practice, the added efficiency in exploring the design space also allows improving performance to levels that would not realistically be possible without anticipative score estimations.

4.1.1.1 Guiding Model Search

Optimisation of an artificial intelligence system is a process of multiple nested levels. The innermost level, where applicable, is a machine driven training loop, e.g. training neural networks with gradient descent at every batch until the end of training. There is at least one level above: developers choosing an optimal design for the problem at hand, e.g. tuning hyper-parameters, selecting architectures or finding optimal training data. This second level of optimisation can –usually only in part– can also be delegated to computational techniques, for example Bayesian optimisation (Frazier, 2018), Neural Architecture Search (Elsken et al., 2019), AutoML (X. He et al., 2021), or meta-learning techniques. Especially when computational techniques are used, this problem is also referred to as bi-level or multi-level optimisation (Franceschi et al., 2018).

We want to avoid doing a grid search over the full design space, training all the corresponding subjects, and then comparing their performance. Developers optimise performance using a certain mental model of performance, knowing which architectures are relevant, which hyper-parameters ranges are sensible. The search space is constrained and the search strategy guided by a performance expectation for every part of the design space, e.g. selecting those designs with the highest expected performance and updating beliefs as evaluation records come in.

Some of the computational techniques also use an explicit model of performance to inform a selection or search algorithm. Bayesian optimisation (Frazier, 2018) is a clear example. It starts off with a human-informed prior distribution $s(r|\mathbf{m})$, where $\mathbf{m}$ is typically a set of hyper-parameters, and maintains a posterior distribution $s(r|\mathbf{m}, \dot{R})$ as evaluation records $\dot{R}$ are collected for different values of $\mathbf{m}$. These distributions act as subject-generalising score predictors and are used to construct what is called the *acquisition function* or *infill sampling criteria*, which selects the designs that are most interesting to collect scores for balancing exploration and exploitation. Figure 4.1 visualises a hypothetical example, and X. Song, Zhang, et al. (2024) is great resource for a concrete implementation that has been implemented and used at Google scale.

Definitely not all bi-level optimisers rely on a score predicting model. For example many meta-learners skip predicting scores and predict hyper-parameters directly,

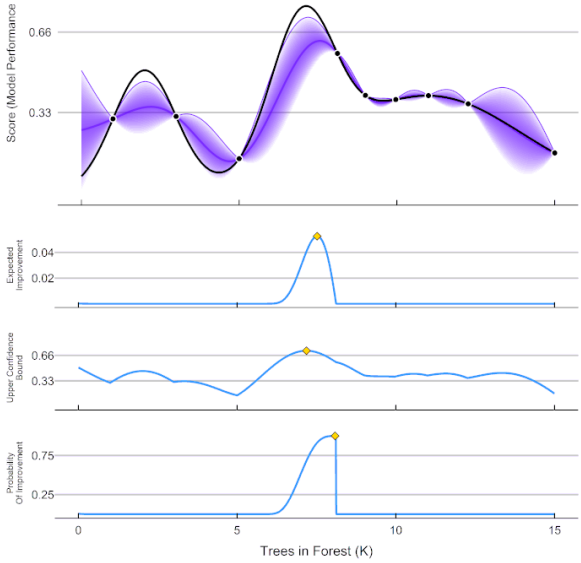


Figure 4.1: Visualisation of Bayesian optimisation. Excerpt from a Bayesian optimisation of the unknown performance (black line) of a random forest as a function of number of trees in the forest (x-axis). The posterior is represented by purple while the black dots represent the observed performance, and the three blue lines represent different acquisition functions. By AnotherSamWilson, CC BY-SA 4.0, no changes made.

or optimise hyper-parameters with gradient descent, evolutionary algorithms[1] or non-linear programming (Hospedales et al., 2020). Although we can see a gradient as making exactly one prediction –"over there will be a better subject for the batch we have just seen"–, this would stretch the definition of a score predicting model a bit far, and is not quite what we care about.

But ideas relying on explicit performance models are common and are spread throughout the literature. For example C. Yang et al. (2019) uses an inner-generalisation score predictor in an AutoML setting. Using a large collection of `(subject, dataset, aggregate score)` triples, when a new dataset is encountered some cheap subjects are trained, and matrix factorisation (section 4.2.2.1) is used to populate the cells for other subjects, after which an optimal subject design can be selected for training.

[1]Evolutionary algorithms can also use score predictors. They are galled *surrogate fitness functions*, and Jin (2011) is broad scoped overview, while Y. Sun et al. (2020) is a concrete application in ML optimisation.

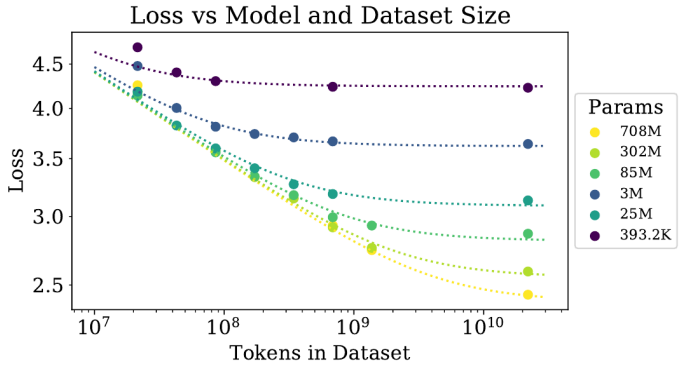


Figure 4.2: Example of scaling analysis. Taken from Kaplan et al. (2020).

Explicit performance models are also common in meta-learning and (neural) architecture search. Sections 6.1.3 to 6.1.5 in Grąbczewski (2014) provide a nice overview of score predicting meta-learners for decision trees; and B. Deng et al. (2017), Istrate et al. (2019), Kokiopoulou et al. (2020), C. Liu et al. (2018), and Y. Xu et al. (2021) are all examples of architecture search relying on explicit score predictors, commonly referred to as *surrogate evaluators*. Lastly, Cheng et al. (2020) deserves special mention, as it does architecture search by doing instance level performance predictions!

Meta-learning, optimisation, AutoML and architecture search are all closely related fields of study, and their respective taxonomies often include or mention each other. What unites them is their need and supply of generalisation in predicting scores at the aggregate level for subjects yet untrained.

4.1.1.2 Guiding Model Scaling

Scaling analysis investigates how subject performance relates to certain scalable design choices such as dataset size or number of parameters. It is a specific and relatively manual variant of guiding model search (as discussed above), but it is a powerful demonstration of the benefits of having a good model of performance. The idea has been in use for while, see e.g. Frey and Fisher (1999), but have attracted a lot of attention recently in the context of training large language or multi-modal models, where multi-million dollar training runs are no longer an exception.

We are particularly interested in the cases where bigger subjects are expected to perform better. The idea is that performance of a subject at a smaller size, but with an otherwise identical design, is predictive of the performance of the larger variants. Concretely, scaling analysis fits a parametric function –often referred to as the scaling law– to the evaluation records of progressively larger but still small subjects. The resulting scaling function allows extrapolation to larger subjects by projecting the slope of the measured performance forward on the scale axis. Typically, training loss or validation loss is predicted on the pre-training dataset, which is itself taken to be indicative of downstream metrics of interest, although e.g. Caballero et al. (2022) predict downstream metrics directly. Figure 4.2 provides an example, and more details and caveats are discussed in section 4.2.3.1.

Scaling analysis allows developers to experiment with the design at smaller scales, where subjects are cheaper and faster to train, and to select an optimal design before committing to the training of a single large variant. As performance keeps increasing with scale, a single large training run can be made to exhaust the full budget in order to maximise performance.

As in Frey and Fisher (1999) and Jain et al. (2023), scaling analysis can also be used to anticipate the quantity of data needed to reach a certain performance threshold, and plan data collection accordingly. And as in Owen (2024), it can also be used to forecast AI performance, not for a specific design, but for the field of AI in general.

4.1.1.3 Designing Curricula

Soviany et al. (2022) writes "Training machine learning models in a meaningful order, from the easy samples to the hard ones ... can provide performance improvements over the standard training approach based on random data shuffling, without any additional computational costs. Curriculum learning (CL) strategies have been successfully employed in all areas of machine learning, in a wide range of tasks."

The key requirement being a determination of easy samples or hard samples, i.e. those where scores are expected to be low or high respectively. As in the previous sections on bi-level optimisation (section 4.1.1.1), not all curriculum learners use an explicit score model, but many –including the traditional approaches– do. Following the taxonomy of Soviany et al. (2022), the methods of interest are largely found in "vanilla CL" and "student-teacher CL". Vanilla CL largely relies on a priori determination of instance difficulty; pg. 6 provides an illustrative collection of difficulty functions and Meng et al. (2024) is an interesting recent method

example. Student-teacher CL "splits the training into two tasks, a model that learns the principal task (student/subject) and an auxiliary model (teacher/score predictor) that determines the optimal learning". T.-H. Kim and Choi (2018) is good example where the teacher method is based on an explicit score predictor.

Training curricula can also be enhanced by curating instances instead of ordering them. For example, active learners are a class of machine learning algorithms that can interactively query another information source (usually humans) to label new data points. Most active learning methods are non-anticipative, see e.g. Settles (2010), so we will not discuss them further. But e.g. Bronstein et al. (2022) use a learned difficulty estimator for upsampling more difficult samples to learn from.

However, for sequential decision problems with procedurally generated environments (Risi & Togelius, 2020), curricula look different: supervision is cheap, since a single reward function can supervise different instances of the task, and new instances can be generated at whim, but the matter at hand is to curate the generated instances in search of those that are of adequate difficulty. The general framework consists of (i) a procedural environment generator, (ii) a predictor that can anticipate a subjects score for a generated environment, (iii) a search and selection algorithm that selects instances for the curricula. Florensa et al. (2018) and V. Bhatt et al. (2022) are exemplary instantiations of this paradigm, with simple assessor-like supervised score predictors. Florensa et al. (2018) introduce generative adversarial network (GAN) to generate goals, where the discriminator of the GAN is a score predicting model. On the other hand Dennis et al. (2020), Gur et al. (2021), and Parker-Holder et al. (2022) merge step (i) to (iii) into a learned generator, which is trained with reinforcement learning (RL) to maximise *regret*: the difference in score between the current subject and an optimal policy. They all use an actor-critic RL method (Konda & Tsitsiklis, 1999), of which the critic acts as score predictor. Campero et al. (2021), does not maximise regret, but uses similar actor-critic generator with score predictor nested in.

4.1.2 Optimising Evaluation

After development comes evaluation; and again, score prediction comes in handy at this stage of the machine learning pipeline. Not that this is very *meta*. We are using score predictors to optimise other score predictors. The concrete use cases should make everything clear quickly.

4.1.2.1 Adaptive Testing

Adaptive testing is to score prediction as active learning is to general prediction. A way to improve sample efficiency by selecting those instances that are optimal for learning. For that reason it is often called active testing in the machine learning community, e.g. Kossen et al. (2021), but the idea had already been use in psychometrics, where it is called adaptive testing.

Adaptive testing is usually applied in order to determine an aggregate score for a particular trained subject. In the procedure, evaluation records are requested in an iterative fashion: (i) given a possibly empty set of evaluation records, request the next record(s) that would be most informative, this is called the *acquisition function*; and (ii) update the set of evaluation records and the corresponding acquisition function with the newly acquired scores. This is done until some stopping criterion is met, e.g. uncertainty about the aggregate score has dropped to an acceptable level. The idea is that this level is reached with less instances than would be needed for passive (i.e. random) sampling.

The acquisition function is where the score predicting model comes in. The selection of the most informative instances is a function of the subject's expected score for those instances. For example Kossen et al. (2021) samples instances proportionally to the expected magnitude of the corresponding error, and psychometrics has developed a suite of acquisition functions for Item Response Theory based score prediction (section 4.2.2.2).

Adaptive testing is often used to specifically optimise the cost of grading, e.g. collecting as little gold labels as possible for a currently unlabelled test set. In that case, there is no need to be anticipative w.r.t. to subject-instance behaviour. The acquisition function could even be based on the uncertainty of the subject. Therefore, non-anticipative score estimation is the norm, see e.g. Ha et al. (2021) and Kossen et al. (2021, 2022).

But the benefits of full anticipation grow as the space of instances over which to search grows and as the cost of observing subject-instance interaction increases. For example when evaluating humans, we preferably give them 10 test questions rather than 100 if the aggregate score is confidently the same. In the case of e.g. robotics, we want to avoid testing with difficult instances too quickly, to avoid the robot damaging itself or its environment.

4.1.2.2 Cost-Effective Evaluation

Related to the above is the notion of (statically) reducing the size of the test set to make evaluation cheaper, while maintaining an accurate estimate of the subject's performance. For example, large multi-task LLM benchmarks such as BIG-Bench (Srivastava et al., 2022) or HELM (Liang et al., 2022) have 'lite' versions with a reduced number of tasks to reduce the resource burden of evaluation. The tasks are selected with the viewpoint that they are equally –or more– predictive of performance on downstream use-cases of the LLM. We care more about an LLMs coding or summarisation ability than its chess playing ability, because the former correlated more with performance on tasks that LLMs are actually used for.

This selection of tasks is based on domain knowledge, i.e. models of performance from expert humans. But we are more interested in algorithmic techniques. For example, through the use of IRT methods (section 4.2.2.2), Polo et al. (2024) created tiny versions of several popular NLP benchmarks, reducing test set size from several thousand instances to hundred or less[2]. They perform second-order evaluation, and find that accuracy measured on their tiny benchmark predicts accuracy on the full benchmark with an average error of less than 2%. Pacchiardi et al. (2024) has the same goal, focusing on HELM Lite, but uses assessor based methods instead, while MixEval (Ni et al., 2024) focuses on predicting performance on the expensive to operate and user-facing ChatBotArena (Chiang et al., 2024) by using a mix of instances from across a wide range of benchmarks, although they only report rank correlation and not absolute performance.

These techniques should see increasing relevance as the scope of evaluation continues to increase and encompass more and more tasks.

4.1.2.3 Removing Contamination

Datamodels (Ilyas et al., 2022) are an exotic score prediction technique that predict subject score for a singular fixed target instance t, using the presence of each training example in the training set as subject features. In other words, if the training set contains $n = 50.000$ instances, $\mathbf{m}$ is 50.000 length binary vector, where each entry indicates whether that sample was used for training. Generalisation is of course needed, as the space of possible vectors is sizeable to say the least. Evaluation records are sourced from 300.000 training runs. A collection of datamodels –one for each instance in the test set– can be used to predict aggregate performance.

[2]It is however not entirely clear from the paper how much improvement this provides over randomly selecting 100 instances.

Section 4.2 in Ilyas et al. (2022) describes the use of datamodels for the removal of data contamination. To paraphrase, datamodels are linear models, and since they are linear models, the magnitude of the parameters $\theta_1 ... \theta_n$ of the datamodel can be interpreted as feature importance of each input feature $m_1 ... m_n$. And since each feature m_i represents the presence of a particular example i during training, the highest-magnitude parameters θ_i reflect the training examples i whose presence (or absence) is most predictive of subject performance. After inspection, Ilyas et al. (2022) reports those instances to be very similar, possibly indicating train test contamination.

4.1.3 Optimising Inference

The most traditional application of evaluation is optimising inference. Is the subject good enough to deploy? Which subject out of several trained ones would be the best? Evaluation here assumes a trained and available subject, and constructs tests to anticipate performance in deployment, where the decision to deploy is coarse, in that one either deploys or not, and if deployed, the subject is used for several instances.

The general principle is to use aggregate metrics and rely on human knowledge for generalisation. In other words, it is humans that assess the likelihood of distribution shift, what the expected range performance is, and whether that is sufficient. We will not dwell on this much, only briefly discussing it in (section 4.1.3.1). Instead we will focus on applications for machine score predictors and applications differing from this default setup, e.g. by being instance focused.

At the instance-level, the advantages of being anticipative focus on saving compute cost and inference time for single input/output prediction problems (e.g. image classification), while sequential decision problems need anticipation to guarantee safety and performance as well.

4.1.3.1 Pre-deployment Model Selection and Rejection

The notion of selecting or rejecting subjects for a new task has seen a lot of research under the names unsupervised evaluation, semi-supervised evaluation, and distribution shift estimation. The setup here usually assumes a sample of instances from the new task, which are unlabelled (and thus unscored). While this is a potential application for score prediction, most methods are –sensibly so– based on self-assessment of the subject, or at least observe subject behaviour first, e.g. Y. Jiang et al. (2022),Baek et al. (2022) or Schelter et al. (2020).

But there are exceptions. For example Elsahar and Gallé (2019) propose several measures of distribution shift, and for a collection of datasets, plot the relation between distribution shift and performance drops. They then regress over these samples, providing a score predictor for arbitrary levels of distribution shift. AndJang et al. (2023) introduce Retrieval-of-Experts. They build an 'expert library', which is a collection of embeddings of representative training samples from a variety of different expert models. When confronted with a new task, they match instances from this new task to those in the expert library, and retrieve the expert whose training samples match most closely. W. Deng et al. (2021) propose a semi-anticipative method. They find that image classification accuracy correlates with the ability to quantify the rotation of an image, something for which labels can be generated automatically. Reusing the backbone of the subject, they train surrogate model on the rotation detection task, and use that performance as an estimate of real task performance.

We expect that anticipative score prediction would provide a more unique advantage if we can not sample instances from the new task, and can therefore not rely on self-assessment or observing subject behaviour.

4.1.3.2 Instance-Level Model Selection (Routing)

Instance-level model selection is picking the best tool for the job. The idea has old roots in traditional computer science (read: non-machine learning), where algorithm selection has been studied for a while (Kerschke et al., 2019; Rice, 1976). The idea is that instances are routed to an optimal subject, based on instance features and subject strengths, taking into account performance, cost, and speed, and aiming to optimise resources, performance, or user experience. When costs are negligible, there is little incentive to be anticipative, and techniques like ensembling (Kuncheva, 2014) or cascading (Streeter, 2018) are used instead when multiple subjects are available[3]. But anticipative score prediction has gained significant traction in the use of large language models, of which a plethora of cloud-hosted variants are simultaneously available through APIs and for which significant diversity exists in performance area, cost or speed[4]. Commercial routing solutions, e.g. Martian AI[5] , or OpenRouter[6], are capitalising on that diversity, which allows –and is needed– for the router to create benefits over

[3]Ensembling and score prediction can also be combined, as in L. Chen et al. (2022), which optimises costs by being anticipative, but still selects multiple subjects to ensemble.

[4]Meding et al. (2021) is an interesting study w.r.t. to score diversity in image classification, showing that 60% of images in the ImageNet (J. Deng et al., 2009) validation set are correctly or wrongly classified by all subjects.

[5]https://withmartian.com/

[6]https://openrouter.ai/models/openrouter/aut

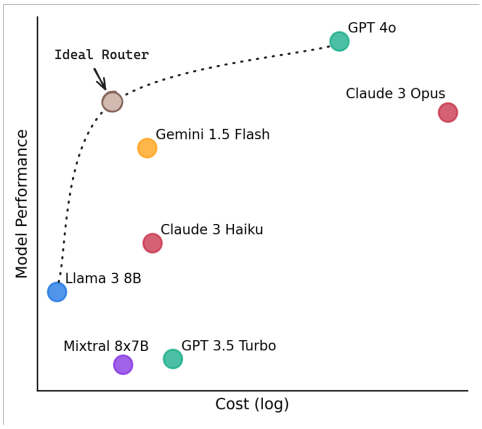


Figure 4.3: Illustrative example of instance-level model routing. The goal is to jointly optimise cost and performance by routing instances to different LLMs. From https://lmsys.org/blog/2024-07-01-routellm/, with associated publication Ong et al. (2024).

using a single subject. Hu et al. (2024) is a benchmark dedicated to the routing use case.

While OpenRouter mentions the use of (undisclosed) heuristics, most of the methods reported in the literature are based on assessors (section 4.2.5.3), albeit with a focus on instance generalisation, as is needed for routing. Examples include C.-H. Lee et al. (2024) and Shnitzer et al. (2023) using a nearest neighbour based assessor, (Nguyen et al., 2024) using contextual multi-armed bandits (based on assessors), and Ong et al. (2024) compare multiple different variants, including a BERT (Devlin et al., 2019) based assessor.

Various applications are also focused on the situation where the number of subjects to route between is exactly equal to two, typically considering a cheaper worse one, and a more expensive superior one. In traditional machine-learning this is called deferral, while in LLM routing this is called hybrid inference (Ding et al., 2023; Ong et al., 2024), and edge computing, this is a part of what is called split computing, where computing is divided between a small edge device and a powerful cloud server. Kag et al. (2022) uses assessors to optimise compute latency for edge inference deferral.

We want to reiterate that for sequential decision problems, instance-level model selection is not just a matter of optimising compute cost and inference time, but of optimising performance in general due to the large class effects that can be

induced by acting in the physical and digital world, e.g. those related to safety. We have not found concrete demonstrations, but they are poised to arrive as agentic AI systems develop wide-spread usage (Shavit et al., 2023).

A preliminary example is Y. Zhao et al. (2024), where assessor models are used to select an an optimal composition for different multi-agent RL settings. This idea could apply to human-AI collaborations as well, and in general, in an ecosystem of interacting agents, tools, and humans, score prediction will be needed to select the entity which to ask for assistance, as humans currently already do on a daily basis. Erdogan et al. (2024) is such an example, where a digital agent has a routing function deciding which external tool to use[7].

4.1.3.3 Conditional Computing

Instead of choosing between different discrete subjects, one can also choose between parameter subsets of a single subject, e.g. sub-networks of single neural network to activate. To make this work, as with selection of discrete subjects, a router model is trained, also called a gating function, which similarly functions as an explicit score predictor. The boundary between what constitutes a single subject versus a collection of subjects is up for interpretation, but *the key difference is that the router is now trained jointly with or as part of the subject*: it observes the scores used as feedback during training, rather than those used for measurement during evaluation on a held out set. This highlights how fuzzy the boundary between learning and evaluating really is[8].

The idea is most commonly referred to as Mixture-of-Expert (MoE) models, but also as sparse experts models (Fedus et al., 2022) or conditional computation (Bengio et al., 2016), the latter of which makes the connection to "adaptive" or "dynamic" computation more obvious (Han et al., 2022; C. Xu & McAuley, 2023). It is commonly accepted that Jacobs et al. (1991) is the first reported use of MoE models, using them to route between different neural networks. Since then, they have been applied to partition the input space between different Support Vector Machines (Lima et al., 2007) and Gaussian Processes (Meeds & Osindero, 2005) as well. Yuksel et al. (2012) provides a good overview of MoE before the era of deep learning, while Fedus et al. (2022) covers various deep neural network and transformer based approaches.

[7] Although it should be noted that the router is trained in supervised way on (instance, tool) pairs, and thus does not model scores.

[8] Yuksel et al. (2012, pg. 2) even sees conditional computation methods as decision trees with soft boundaries, something that becomes more obvious when considering multi-level routers (pg. 3), and reminds us of higher-level evaluation (section 3.2.2).

Routing techniques are varied and diverse. For example Bengio et al. (2016) uses reinforcement learning to learn the router, having recast the problem as a Bandit problem, while the influential Shazeer et al. (2017) make the router differential together with the rest of the subject. A related concept, known as policy networks, is also relevant in this discussion. For a comprehensive exposition, see Han et al. (2022).

A complete overview is beyond the scope of this thesis, but several key points are worth noting. Firstly, multiple experts might be selected simultaneously, rather than relying on a single one. This requires careful partitioning of the input space, known as load balancing, to avoid overloading a few experts. In addition, sub-parts of instances, such as individual words in a NLP task, can be routed to different experts In the context of our framework, routers make relative score predictions, usually assigning a weight to each expert; the top-k experts are then selected based on these weights. Importantly, all routers generalise across instances, ensuring robust performance across multiple input scenarios.

4.1.3.4 Instance-Level Rejection

Instance-level rejection refers to the idea of simply not interacting with a subject when the expected performance is too low.

In supervised learning the only reason to do this preemptively should be to optimise compute cost and inference time, as we do in L. Zhou et al. (2022), where we predict the performance of a large GPT model (Ouyang et al., 2022) on a data-wrangling task with a small random forest and manually crafted features. But anticipate rejection is nonetheless common, for the reason that observing subject behaviour does not help. See NannyML (2024) and Drapal et al. (2024) for examples, but especially Hendrickx et al. (2024, Sec. 8), which provides an excellent overview of machine learning with a reject option with external and anticipate reject models (i.e. score predictors).

As with instance-level model selection, sequential decision problems form a more pertinent need for anticipative score prediction. There is an active research area for rejecting reinforcement learning subjects –or more precisely, their actions– under the name of 'shielding'. Especially Model Predictive Shielding (Bastani, 2021), Dynamic Shielding (Waga et al., 2022), and Online Shielding (Könighofer et al., 2023) are techniques that fit our notion of explicit score predictors.

These techniques have mostly been applied to single-task reinforcement learning, where there is single and specific environment in which to do optimal control.

As multi-task open environments, e.g. computer control (Xie et al., 2024) or as general household assistance (Brohan et al., 2023) are becoming more feasibly with the advent of GPAI, shields and related techniques are poised to become more prominent.

4.1.4 Predicting for Prediction's Sake

Apart from control, we sometimes value prediction for epistemological reasons. In the applications described below, performance could be measured instead of predicted, but it would defeat the point. We are specifically interested in the how, what, and why of score prediction; we are not interested in the material effects, it is the model of performance –the information connecting instance, subject, and outcome– that is of interest.

4.1.4.1 Learning from Predictive Models

All the previous sections have been about learning, of course, but here we are specifically concerned with transferring and extracting knowledge from machine score predictors to humans. In the science and practice of artificial intelligence, we want –or at least we need to– understand what makes subjects capable and why certain problems are hard for machine cognition. Much of this knowledge comes from the general scientific process, of course, but as the scope of evaluation has increased (section 2.6), we may need the help of machine intelligence to help us understand the myriad of tasks where AI is now performing at a level that is not trivial, but definitely not perfect either. For example it is hard for humans to analyse terabyte sized datasets, billion parameter models, or data formats that were never quite interpretable to humans to start with (e.g. those in protein folding). So to cheaply repeat a quote from the introduction, "using complex predictors may be unpleasant, but the soundest path is to go for predictive accuracy first, then try to understand why" Breiman (2001). We are interested in the mechanisation of error-analysis; to find the patterns in performance, if there are any to be found.

There is no singular area of research encompassing this idea, so we will highlight a few selected techniques where interpretability or explainability methods were used to transfer score predicting knowledge from machine to human. First up is a classic: Shapley values (Merrick & Taly, 2020). Shapley values are a theoretically grounded method for measuring influence of features to outcomes. This can be done either *globally*, meaning over the entire dataset, or *locally*, meaning for a particular instance and prediction. There are three different angles to explore:

(i) directly explain which instance features are important to the subject loss, (ii) directly explain which subject features are important to the subject loss, and (iii) explain which features are important for a score predicting model.

First, we discuss the angle of calculating feature importance of instance features with respect to subject loss. While Shapley values are commonly used to calculate feature influence on subject *behaviour*, calculating feature influence on subject *loss* is an equally natural use (Grömping, 2007, 2009; Loecher et al., 2022), with little to no difference in calculation. In particular, since many of the methods used are approximations of Shapley values, e.g. G-DeepShap (H. Chen et al., 2022), Shapley values can be considered as score predictors with generalisation, mapping from subsets of instance features[9] to (relative) scores, and explaining to humans which features make prediction easier.

Secondly, Data Shapley (Ghorbani & Zou, 2019; Jia et al., 2019) is an idea that is closely related to Datamodels (section 4.1.2.3) and curriculum learning (section 4.1.1.3). It assigns Shapley values to each training instance, quantifying the contribution of each instance to the subject performance. Because there is exponential complexity in the number of data points, Data Shapley values are always approximated, and as such turn into turn into generalising score predictors. They consider the exact composition of the training data as the subject representation, and explain to humans which instances improve aggregate scores and thus facilitate learning.

Thirdly, instead of doing all generalisation in the calculation of the Shapley values, it is also possible to train a regular score predictor, and calculate Shapley values for the score predictor (w.r.t. subject features, instance features, or both). Q. Zhang et al. (2024, Section 5.3) is such an example, where they build a score predictor for use in scaling analysis, and do a Shapley analysis over both item features, including task family and number of shots, and subject features, include data size, parameter count, model family, batch size, context window and much more. They find for example that –for the subjects under evaluation– dataset size is significantly more important then parameter count.

Once a score predictor is created, we are not limited to simply using Shapley values. Any explainability technique can be used. Prudêncio et al. (2024) for example builds a score predictor for two tabular medical datasets, dealing with diabetes and COVID respectively, and after identifying important features, also displays 1-way and 2-way partial dependence plots for the most important features.

[9]Confusingly, from the perspective of a score predictor, which instance features are used to train a subject could reasonably be thought of either as description of the problem or of the subject

4.1.4.2 Fortifying Theory

According to the philosophers of science discussed in the introduction, every time we make prediction about performance, and that prediction later turns out to be true, we are confirming the theory and the understanding that led to that prediction. On the flip side, if a commonly accepted prediction does not hold, our understanding will undergo some sort of change or revolution (T. S. Kuhn, 1996).

For example, every time a certain scaling law turns out to be (in)accurate, we confirm or reject the theory behind it. Coincidentally, the scaling laws have already proven to be limited or wrong in several iterations: Kaplan et al. (2020) provided an initial calculation of the optimal balance of parameter count and dataset size for LLMs under a given compute budget, but Hoffmann et al. (2022) found that prediction not to hold, instead requiring models to be smaller but trained with more data. Touvron, Lavril, et al. (2023) and Touvron, Martin, et al. (2023) then found that the proportion of the dataset size in the compute budget was again underestimated, partly because it failed to integrate the compute cost during inference. This sequence of revisions shows both that our understanding of scaling behaviour was quite limited in its exact form, but also repeatedly confirmed the general notion of scaling laws. While the exact numbers might differ and do not transfer well between (Schaeffer et al., 2024), for a specific architecture and data setup, it is possible to fit a scaling curve, and extrapolate performance accurately.

Similar principles apply every time data contamination is removed, e.g. according to section 4.1.2.3, and we find performance to drop significantly, or when error-analysis in one set of experiments explains the scores in another. The scientific understanding of subject capabilities depends on the continued making of new predictions.

4.1.4.3 Legal liability

Imagine that a car breaks down during operation. Suppose the breaks fail and the car then hits a pedestrian crossing the road, possibly causing death or serious injury. The question of who is responsible for the harm caused, be it the manufacturer, the driver, or maybe even the pedestrian, is the topic of study in that part of the legal process called "liability law", as the responsible party is liable to provide compensation for the caused harm. It could be that there was a production mistake, and the car did not match the standard of safety. It could be that the driver –assuming they are also the owner of the car– failed to maintain the integrity of the vehicle, not completing regular check-ups that would be expected

for a braking system. It could be that the driver simply drove too fast, or that the pedestrian crossed the street where or when they should not have, and an accident would have happened regardless of the brake failure. To determine who, if anyone, is at fault, all these questions need to be answered.

There are many different concepts and areas of liability law: fault-based liability, risk-based liability, product liability, standards of care or standards of safety. Some parts are covered by insurance, some by consumer-contractual law. A comprehensive understanding of this vast area of study is definitely out of scope, but for the interested reader, Llorca et al. (2023) provides an interesting overview and discussion of liability with respect to AI systems. What is of interest to us, is the principles that return again and again: foreseeable error, expectation of performance, expectations of safety. Even if we can observe the behaviour, and in fact we have, including the harm it caused, it is in a way, mostly ignored. It is scored and graded, yes, but the reasoning is counterfactual: could the respective party, before the harm was caused, have foreseen and avoided the harm their later actions would cause. This is of course, a question of performance prediction. It is one that can be answered by proper evaluation, and for which a failure to conduct proper evaluation is a matter of negligence.

In terms of AI systems, relevant factors can be whether a representative test set was used, whether distribution shift should have been expected, whether evaluation results were sufficiently granular or whether foreseeable demographic bias was present. As usage and deployment of AI systems increases, a proper model of their performance –and Llorca et al. (2023) specifically cite unpredictability as a complicating factor in liability law– seems an essential requirement for sensible insurance contracts, reasonable consumer-contractual law, and protecting the human rights codified in the legal process.

4.2 Methods

Now that we have discussed the applications of score prediction, we can present an overview of the methods of score prediction. Some of these methods might not be conventionally associated with evaluation, e.g. because there is no human in the loop, but they all consistently build a model of performance, aiming to predict scores for new instance or subjects.

We will discuss over 15 different score predicting methods, including factor analysis (section 4.2.2.1), scaling laws (section 4.2.3.1), intrinsic difficulty functions (section 4.2.4.2), and capability oriented evaluation (section 4.2.5.4). Section 4.3 will discuss in depth how various overarching properties can relate and differenti-

ate between these methods, but to help make sense of the structure in the current section, we present a sneak peek here already.

The most important properties are the *granularity* of the prediction, the *kind of information observed* about the subject and instance, and the *kind of generalisation*. More straightforward, the granularity of the prediction simply refers to whether a score prediction can predicts scores for a singular instance, or instead predicts for collection of them. We will use the term *item* when it does not matter what the granularity of the prediction is.

Regarding the type of information observed, there are three options for both instance and subject: we can simply observe nothing about the instance or the subject, ignoring it completely (usually not both at a time); we can observe a unique identifier, denoted m_{id} and x_{id}; or we can observe a representation $\mathbf{m}$ or $\mathbf{x}$ of machine-readable features. Specifically for instances, an additional commonly observed piece of information is the task id, an identifier representing the assumed distribution an instance is sampled from. To analyse what information is observed, we consider the following question: given a collection of evaluation records, containing scores from multiple subjects across multiple instances, potentially from different tasks, what columns and features does the learning algorithm of the score predicting model look at?

Lastly, scores are always predicted for a specific combination of subject and item. This composite input space entails five different kinds of generalisation:

1. no generalisation, where the score can not make score predictions for a specific `(subject, item)` unless scores have been observed for that exact pair

2. inner generalisation, where the score predicting method can make predictions for an unseen `(subject, item)` combination only when it has observed scores for both the subject on other items, and scores for the item from other subjects

3. subject generalisation, where the score predicting method can make predictions for an unseen `(subject, item)` combination when it has observed scores for the item from other subjects;

4. item generalisation, where the score predicting method can make predictions for an unseen `(subject, item)` combinations when it has observed scores for the subject on other items

5. and lastly, full generalisation, where the score predicting method can make predictions for an unseen `(subject, item)` pair while scores from neither the subject nor the item have been observed.

These generalisation types will also serve as the grouping for the methods in this section, starting with evaluation techniques that do not provide generalisation. Table 4.1 provides an overview of the methods we will discuss.

Generalisation	Method	Inst. Info.	Subject Info.	Granularity
No	Performance Metrics	Task ID	No	Dataset
No	Populational Difficulty Metrics	No	Popul. ID	Instance
Inner	Factor Analysis	ID	ID	Any
Inner	Matrix Factorisation	ID	ID	Any
Inner	Item Response Theory	ID	ID	Any
Subject	Scaling Laws	Task ID	Features	Dataset
Subject	Bayesian Optimisation	Task ID	Features	Dataset
Subject	Subject Assessors	Task ID	Features	Dataset
Item	Intrinsic Difficulty Measures	Features	None	Instance
Item	Agent Characteristic Curves and Grids	Features	ID	Instance
Item	Anomaly Detectors	Features	ID	Instance
Full	Assessors	Features	Features	Instance
Full	Aggregate Assessors	Features	Features	Dataset
Full	Measurement Layouts	Features	Features	Instance
Full	Human Mental Models	Features	Features	Any

Table 4.1: Organisation of the various score predicting methods.

4.2.1 No generalisation

Evaluation techniques that do not generalise cannot make predictions for unseen `(subject, item)` combinations. If we would use a no-generalisation method to predict for unseen pairs, all predictions would be identical. The predictive aspect – i.e. what differentiates the method from a lookup table– lies in the fact that scores can be stochastic for a given (subject, item) pair (see section 4.3.3.1) combined with the fact scores are usually a finite-data estimation of an infinite-data notion of performance.

To paraphrase the famous definition of machine learning by Mitchell (1997), 'a computer program is said to learn from experience if its performance at the task improves with experience'. So while methods without generalisation are simple summary statistics, not conditioned on item or subject, their accuracy improves with more evaluation data, matching this delimiter.

4.2.1.1 Performance Metrics

Generalisation	Granularity	Subject Info	Instance Info
None	Dataset	ID	Task ID

As already discussed in section 2.4.1, the most common procedure for evaluating a subject is to hold out some test instances from the collected dataset, on which not to train the subject, and with which the empirical risk is calculated. Cross-validation techniques (Browne, 2000) can be used to estimation of the risk, especially if data is scarce, but can incur a significant training overhead, as the subject needs to be re-trained several times with different train-test splits.

For the empirical risk to be an accurate estimation of the generalisation risk during deployment, the test set must be representative of the deployment distribution. Metrics by themselves therefore do not generalise: they cannot make statements about the performance on an (unscored) dataset with a potentially different distribution.

> **A saving grace.** Metrics can be used to make predictions with non-zero refinement for new instances by predicting the average score for the task the instance belongs to, assuming scores for that task have been observed. But this requires inferring the task to which the instance belongs, which is something the metric itself can not do, and for which we rely on a meta-level information source, e.g. humans.

4.2.1.2 Populational Difficulty Metrics

Generalisation	Granularity	Subject Info	Instance Info
None	Any	Population ID	ID

As explained in fig. 3.2, populational difficulty measures are the antithesis of aggregate performance measures. Instead of aggregating a subject's scores over instances, they plainly aggregate an instance's score over subjects.

For a populational difficulty to be an accurate of estimation of a subjects score, the subject must come from the same population as was used to determine the difficulty. But that population can be quite wide, e.g. all trained subjects of particular model family, and in general we do expect diversity in the subjects capability for a population. That way, even without generalisation, populational difficulty can be used for e.g. adaptive testing (Varshney et al., 2022), finding

mislabelled data (Swayamdipta et al., 2020), or designing curricula (X. Zhang et al., 2018).[10]

The aggregation over subjects must not necessarily be an average, e.g. Ethayarajh et al. (2022) uses the infimum over a computational family of subjects.

Note that there are many other techniques that also rely on populations to estimates of difficulty, but these, as will be discussed, take into account the subject, rather simply aggregating over them.

4.2.2 Inner Generalisation

Inner generalising score predictors can only predict the score to make predictions for an unseen `(subject, item)` combination if they have observed scores for both the subject on other items, and scores for the item from other subjects.

4.2.2.1 Factor Analysis and Matrix Factorisation

Generalisation	Granularity	Subject Info	Instance Info
Inner	Any	ID	ID

Factor analysis Mulaik (2009) and matrix factorisation Koren et al. (2009) are very related ideas, used in entirely different fields for a variety of different but related purposes.

Factor analysis is a method used in psychometrics to describe the variability among observed, correlated test scores in terms of a potentially smaller number of unobserved variables called *factors*, where factors then typically represent general abilities, e.g. athletic ability or verbal reasoning ability.[11] Some of the main goals are to find out how many latent variables are needed to explain the correlation among observed scores, to analyse to factor strength and direction of the factor 'loadings', which are the correlation coefficient between test scores and the latent variables, and to analyse the general factor structure, i.e. which variables load on which factors. Burnell, Hao, et al. (2023) is an example of factor analysis applied to large language models.

[10] The population size is one here. In other words, the scores of a single subject are assumed to be representative for another.

[11] Apart from as a performance modelling technique, factor analysis is also used in other sciences, e.g. political science or biology, or for other psychometric variables such as personality traits.

Matrix factorisation on the other hand is a class of algorithms used in recommender systems to match users to products, e.g. movies to watch or goods to buy. The idea is similarly to explain the observed user-product interaction behaviours (e.g. views or purchases) with a set of unobserved variables, again called latent factors. The difference is that matrix factorisation techniques are designed to work with large and sparsely populated matrices, as we can not expect that all users have watched all movies, and the goal is specifically to fill the empty cells of the matrix, rather than determine the number of factors or analyse the factor loadings. While recommender systems have nothing to do with evaluation in itself, various developed matrix factorisation techniques have served as inspiration for papers oriented to score prediction, e.g. Schram et al. (2023), C. Yang et al. (2019), and Q. Zhang et al. (2024).

For both fields the general mathematical framework is as follows. For a given score matrix $R \in \mathbb{R}^{|I|\times|M|}$, where M is the set of subjects and I the set of items, the idea is to factor R into $R = CD$, with k being the number of factors, with $D \in \mathbb{R}^{|I|\times k}$ being the item-factor matrix, also known as the loading matrix, and $C \in \mathbb{R}^{k\times|M|}$, being the subject-factor matrix, also known as simply the factor matrix. As the factors represent various sorts of abilities[12], D represents the various demands of each instance, and C represents for the various ability-values for each subject.

For an instance with known demands and a subject with known ability-values, a predicted instance level score then looks like this:

$$\hat{r}_{m,i} = d_{i,1}c_{1,m} + \cdots + d_{i,k}c_{k,m} \tag{4.1}$$

although in factor analysis typically the mean score $\sum_{m\in M} r_{m,i}$ is subtracted from $r_{m,i}$ and a noise term is added.

While factor analysis is not typically used to predict scores for `(subject, item)` pairs, it could be, as the principles are the same. In any case, both techniques are inner generalisers.

[12] At least that is the idea. But if the ability correlates very well with design decisions, e.g. scaling related ones such number of parameters or dataset size, it is worth asking if the factors still represent abilities, or instead just measures which design decisions have effect on which tasks.

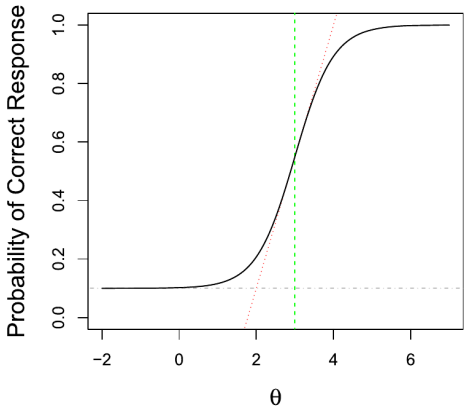


Figure 4.4: Example Item Characteristic Curve. Example of a 3PL IRT model (in black), with slope = 2 (discrimination, in red), location = 3 (difficulty, in green) and guessing parameter = 0.1 (chance, in grey). From Martínez-Plumed et al. (2019).

4.2.2.2 Item Response Theory

Generalisation	Granularity	Subject Info	Instance Info
Inner	Any	ID	ID

Item Response Theory (de Ayala, 2009) is a another method that has its roots in psychometrics. As with factor analysis, it is assumed that there is a latent variable representing an ability. But IRT assumes that there is single latent variable,[13] also called the trait, and usually denoted θ.

The second assumption is that performance can be modelled with an Item Characteristic Curve (ICC), also called the Item Response Function (IRF). The ICC is parametrised function $p_x(\theta; \phi)$ that for a particular item x –e.g. a question– maps the trait θ to to probability of a particular response r, typically a score. From a matrix of `(subject, item)` responses, for each item a the free parameters ϕ_x of the corresponding ICC are fit, as well as the trait-values θ_m for each subject m. Fitting can be done with maximum likelihood estimation, but many techniques exist, and many different functional forms exist for the ICC, dealing e.g. explicitly with binary, categorical, or continuous scores. The 3PL model is a common choice, which is logistic curve with 3 free parameters, representing item difficulty, the slope of the logistic curve –also called discrimination–, and the random guessing chance. Figure 4.4 shows an example ICC.

[13] Whether that assumption holds is something that could be investigated with factor analysis.

Martínez-Plumed et al. (2016) is good introduction to IRT in machine learning, and the following are some further example: Martínez-Plumed et al. (2019) uses IRT to explain why some instances are hard to classify, Zhuang et al. (2023) uses IRT for adaptive testing, Meng et al. (2024) uses IRT for designing curricula, and Polo et al. (2024) uses IRT to construct smaller versions of benchmark datasets that correlate highly with performance on the full set.

4.2.3 Subject Generalisation

Subject generalising methods can make predictions for an unseen `(subject, item)` combination when it has observed scores for the item from other subjects. In other words, for a given and seen item, it can predict performance for unseen subjects, which makes these methods particularly adept at optimising development, helping the search for better subject designs (section 4.1.1.1).

4.2.3.1 Scaling Laws

Generalisation	Granularity	Subject Info	Instance Info
Subject	Aggregate	Features	Task ID

As described in section 4.1.1.2, scaling analysis investigates how subject performance relates with certain scalable design choices such as dataset size or number of parameters.

The idea is that we can analyse a model design ϕ with a set σ of scalable parameters by training a sequence of progressively scaled up subjects $s_{\phi,\sigma_1}, s_{\phi,\sigma_2}, ...$ with $\sigma_i < \sigma_{i+1}$ and extrapolating their performance to larger scales. Extrapolations happens by fitting a parametric function $r = f_\phi(\sigma;\theta)$, where r is some measure of aggregate performance of $s_{\phi,\sigma}$, and θ are the free variables of f, and then checking the projected performance $r^* = f_\phi(\sigma^*)$ at the scale of interest σ^*, for which s_{ϕ,σ^*} has not yet been trained. If we do so for several designs, we can compare which one will have the best performance at point σ^* without training the corresponding expensive subjects.

The function f takes many different forms, but they often are power laws, as these are found to be the best predictors in many scenarios and domains(Hutter, 2021). An example with uni-dimensional σ looks like this:

$$r = f(\sigma; a, b) = a\sigma^{-b},$$

but multiple scaling parameters are also possible, e.g. as in Hoffmann et al. (2022). Instead of power-laws Owen (2024) uses a sigmoid function, and Caballero et al. (2022) uses piecewise power law.

Scaling laws have been used both to predict LLM pre-training loss (Kaplan et al., 2020) and their downstream performance on benchmarks (Owen, 2024). Apart from LLMs, they have also been used in Reinforcement Learning (Hafner et al., 2020), Decision Trees (Frey & Fisher, 1999), and computer vision (Alabdulmohsin et al., 2022).

In general, fitted scaling laws do not transfer well to different subject designs or to different tasks. Predicting downstream performance has also been considered harder than pre-training loss (Schaeffer et al., 2024), and with language models a phenomenon of 'emergent capabilities' has been reported (Wei et al., 2022), where downstream performance sees sudden spikes in performance at a certain scale. Schaeffer et al. (2023) however convincingly argues that this occurs because the measured scores –even at the instance-level– destroy information, e.g. by taking binary correctness instead of probability assigned to the correct response.

> **Extrapolating other curves.** Learning curve analysis is a similar idea to scaling analysis, where instead of scaled design parameters, the x-axis is the number of epochs, or more generally, compute or data spent on training for an already (partially) trained subject. These can also be used for comparing model designs, as in Domhan et al. (2015), usually in more automated fashion than scaling analysis. They are also subject generalisers if we consider the model at different steps during training as different subjects. Hutter (2021) provides a comprehensive overview, including coverage of scale-based curves.

4.2.3.2 Interlude: Scaling the Scaling Laws

Generalisation	Granularity	Subject Info	Instance Info
Subject	Instance	Features	Features

All current scaling analysis is done at the aggregate level, and only takes into account a few subject parameters at a time, noticeably dataset and model size. In Schellaert, Hamon, et al. (2024) we argue that this entails both an *oversummarisation of subjects* and an *oversummarisation of task performance*, and this oversummarisation makes us unable to model realistic performance patterns, causing inaccurate scaling predictions.

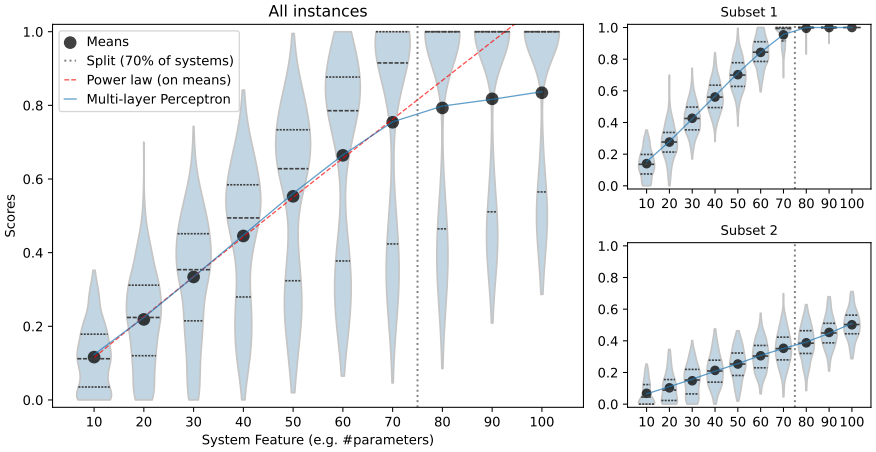


Figure 4.5: Synthetic scaling scenario requiring instance-level analysis. The task performance correlating with subject scale cannot be modelled from aggregate measures, while being completely regular from an instance-level perspective. The plot shows ten synthetic subjects, whose synthetic evaluation scores are designed to be dependent on an abstract feature of the system. For example subject 2 has feature-value 20 (e.g. number of parameters), and has a mean score of about 0.2. The violin plots, with the quantiles marked, represent the distribution of scores of the respective systems. The red line is a power law fitted to the mean scores, while the blue line represents the aggregated predictions of a simple multi-layer perceptron (MLP) that predicts instance-level scores. Both are trained/fitted on the smallest seven systems only. The last three subjects then act as a test for the performance predictor.

Figure 4.5 is such an example, where the dataset is comprised of two subsets of instances with different scaling behaviours. An example ask could be a sentiment classification task of English text, where the domain naturally contains a blend of English varieties, e.g. 'standard English' acting as subpopulation 1, and African-American Vernacular English (AAVE) acting as subpopulation 2. One those two might be more difficult to analyse the sentiment of, and oversummarisation of task performance does not allow us to model this behaviour.

While the example is obviously exaggerated and idealistic, benchmark subgroups are not uncommon (Siddiqui et al., 2022; Swayamdipta et al., 2020). In a high-dimensional problem space like NLP, the subgroups are however not as crisp as in our example, and identifying them is far from straightforward; this complexity is precisely why we need more sophisticated statistical methods beyond simple aggregate measures. In general, it is hard to isolate a single capability in benchmark design (AERA, 2014; Hernández-Orallo (2017)), if that even makes sense for novel kinds of intelligence like LLMs.

While heavy summarisation is sensible in the light of interpretability or data scarcity, it comes at a cost of generalisation and predictive power. With major NLP evaluation efforts like BIG-Bench (Srivastava et al., 2022) and HELM (Liang et al., 2022) producing huge quantities of instance-level evaluation results across a plethora of different AI systems, it is time to capitalise on the available data, and much like we scale AI itself, to also *scale the inferential tools we use in our analysis of AI*. In other words, we propose to use assessor models as the score predictor in scaling analysis.

The other part of our proposal, reducing the summarisation of subjects by integrating more design features, has recently been explored partly by Q. Zhang et al. (2024), who consequently demonstrated accuracy improvements over traditional scaling laws, although they focus on interpolating in a given range of scales between different subject designs.

4.2.3.3 Bayesian Optimisation

Generalisation	Granularity	Subject Info	Instance Info
Subject	Dataset	Features	Task ID

For Bayesian optimisation, we refer back to section 4.1.1.1, but highlight again the following aspects: it is the Bayesian posterior distribution, not the acquisition function, that is the score predictor. The posterior distribution is often

constructed from a supervised learning method that is fit for continual learning, e.g. Gaussian Processes (Seeger, 2004).

In that regard, the posterior distribution could be considered an aggregate assessor (see section 4.2.5.2).

4.2.3.4 Subject Assessors

Generalisation	Granularity	Subject Info	Instance Info
Subject	Dataset	Features	Task ID

With subject assessors we refer to the commonly occurring score predicting models in hyperparameter optimisation and Neural Architecture Search (NAS) that are assessor-like in nature –they are supervised learners predicting scores–, but that predict scores at the dataset level rather than instance-level, and that do not observe instance or dataset features. Like in Bayesian optimisation, they shape the data collection process further, and are often fit for continual learning.

Y. Sun et al. (2020), Tang et al. (2020), and Y. Xu et al. (2021) are example methods, as is B. Deng et al. (2017), which focuses on scaling analysis.

The architecture search algorithms can often be easily modified to take into account dataset features, and can be applied to different problems in general, but the score predictors themselves are tied to a specific problem with a specific dataset. For a different dataset, the score predictor would have to be retrained. Istrate et al. (2019) and Kokiopoulou et al. (2020) are some exceptions, being aware of dataset features, and in this way turning into an AutoML and meta-learning approach (section 4.2.5.1), where the score predictors more often condition on characteristics of the problem.

4.2.4 Item Generalisation

Item generalising methods can make predictions for an unseen `(subject, item)` combinations when they have observed scores for the subject on other items. In other words, for a given subject, it can generalise to unseen items. Item generalisers are used for a myriad of applications, but they are a necessity for optimising inference during deployment (section 4.1.3).

4.2.4.1 Anomaly and Distribution Shift Detectors

Generalisation	Granularity	Subject Info	Instance Info
Item	Instance (Anomaly)	Features + ID	Features + ID
Item	Dataset (Distr. Shift)	Features	Features

Both anomaly detectors and distribution shift detectors capture the same idea: the chance that the subject performs well decreases as the instances (or datasets) become more dissimilar to the instances observed during training. They are novelty detectors, focusing on modelling epistemic uncertainty.

The level of distribution shift for a new dataset can be quantified, and this can be used to place theoretical bounds on the difference in performance, as in Ben-David et al. (2010): if the distribution shifts only a little bit, performance shifts only a little bit as well. Miller et al. (2021) investigates this empirically, and finds these bounds to hold in practice (and often with margin). It should be noted that most techniques for distribution shift estimation rely on (unlabelled) samples from the new distribution being available, and as such, many techniques are not anticipative, instead relying on the self-assessment of the subject. Ben-David et al. (2010) as mentioned above is an exception, specifically their $\mathcal{H}$-divergence.

The same goes for anomaly detectors, which are the instance-level equivalent of distribution shift detectors, and which assess whether a certain instance x is within the distribution represented by the training data. Here, anticipative detectors are more common, see e.g. Markou and Singh (2003), or Hendrickx et al. (2024, Sec. 4).

In any case, these novelty detectors usually do not learn from the subject scores at all. Instead they are based entirely on the observing the training or test set instances x, without the corresponding scores. The training set in this case could be seen as high-information description of the subject, and in that case, the novelty detector could be considered to be a full generalising method.

4.2.4.2 Intrinsic Difficulty Measures

Generalisation	Granularity	Subject Info	Instance Info
Instance	Any	None	Features

With intrinsic difficulty measures we refer to notions of difficulty that are not populational, in that the difficulty can be estimated from observing the item as is, without needing to observe scores first. For example, because we can deduce

that there are more reasoning steps involved in a mathematical word problem, because there is noise in the image to classify, or because a sentence of which to analyse the sentiment contains several negations.

Intrinsic difficulty functions are usually instance-based. Aggregate difficulties definitely also exist, but these are often non-anticipative, as they rely on the labelling of the instances, e.g. to investigate class imbalance. Soviany et al. (2022, Sec. 3) has a good overview of difficulty functions used in the context of curriculum learning. Difficulty estimators do not need to be classical deterministic algorithms, they can also be learned –i.e. machine learning systems themselves– , as in Bronstein et al. (2022) or T. Li et al. (2024). These learned difficulty estimators can learn from populational evaluation data and then generalise to new instances, as in Bronstein et al. (2022) which learns from (instance, score) pairs, or as in T. Li et al. (2024), be learned to annotate the image with various high-level features that are indicative of difficulty, which learns from (instance, annotation).

By themselves, intrinsic difficulty functions only provide relative estimates of performance: instance x_1 being more difficult than instance x_2, which only tells us that the expected score is lower, not what that score will be. That can be sufficient for ranking and selecting instances, e.g. for curriculum design, but not for rejections based on the absolute expected performance. To provide absolute score predictions, intrinsic difficulties are typically combined with agent characteristic curves (see below).

4.2.4.3 Agent Characteristic Curves and Grids

Generalisation	Granularity	Subject Info	Instance Info
Inner (pop. diff.)	Any	ID	ID
Item (intr. diff.)	Any	ID	Features

Agent Characteristic Curves (Hernández-Orallo et al., 2021) are the mirror image of IRT's Item Characteristic Curves. They are a mapping from item difficulty to subject performance, as opposed to the reverse. Agent Characteristic curves can be fitted to empirical evaluation records, and when a new instance is seen (with known difficulty) we can map it to a corresponding level of performance. Fitting a an parametrised function to use as a mapping is not a requirement, binning on difficulty level is also common practice.

A characteristic curve, as in fig. 4.6 assumes a uni-dimensional notion of difficulty. But that does not need to be the case. We could also condition on two dimensions

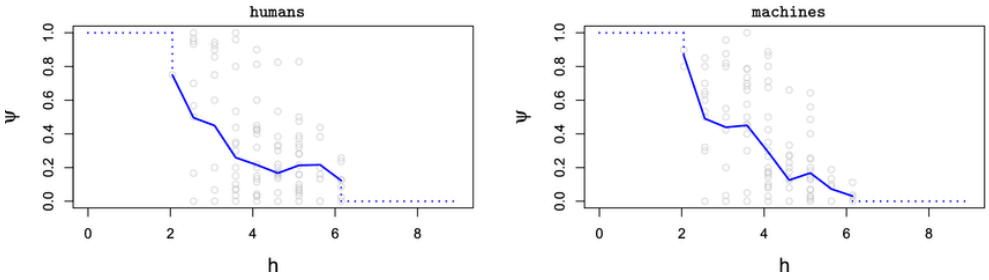


Figure 4.6: Example Agent Characteristic Curve.

of difficulty, or any two features correlation with performance, and get an Agent Characteristic Grid (ACG), as in (Burnell et al., 2022). The binning approach works just as well in higher-dimensions, up to the point where when higher-dimensionality also entails sparsity, i.e. when we cannot reasonably expect to evaluate a subject on multiple instances in every multi-dimensional bin.

At that point, we need to look again at machine learning techniques, e.g. assessors, that generalise over the various difficulty dimension.

4.2.5 Full Generalisation

Score predicting methods that have full generalisation can make predictions for an unseen `(subject, item)` pair while scores from neither the subject nor the item have been observed. There are however only a few actual demonstrations where generalisation happens over both subjects and instances at the same time.

4.2.5.1 A Note on Meta-learning and AutoML

Meta-learning is very related to score prediction. But meta-learning is not simply one technique. It is the general idea, as Vilalta and Drissi (2002) describes, 'to building self-adaptive learners: learning algorithms that improve their bias dynamically through experience by accumulating meta-knowledge'.

As discussed briefly already in section 4.1.1.1, meta-learners are not score predictors. Their primary goal is to meta-learn how to learn various tasks, with the tasks being subject tasks –e.g. image classification or sentiment analysis–, not score predicting tasks.

The two ideas collide in the fact that meta-learning can rely on score predictors, especially to speed up learning. A big class of explicit score predictors can be found in *surrogate fitness functions* (Jin, 2011) for evolutionary algorithms, and these are often used as the algorithm for meta-optimisation (Hospedales et al., 2020), and Grąbczewski (2014, Sec 6.1.3-6.1.5) describes various techniques for meta-learning that rely on explicit score prediction for decision trees, which often translate to other styles of subjects.

But the these explicit score predictors used in meta-learning fall in other categories already discussed, e.g. they are Bayesian optimisers or assessors. In that regard, meta-learning is more of an application than a method. For AutoML, the reasoning is exactly the same.

> We also need to differentiate between meta-learning techniques that rely on score predictors, such as discussed above, and meta-learning techniques for score predictors. Since score predictors are also machine learning systems, they can also be improved by meta-learning. Kokiopoulou et al. (2020), Jain et al. (2023), Shnitzer et al. (2023) and Drapal et al. (2024) are all examples.

4.2.5.2 Aggregate Assessors

Assessors are instance-level by definition (section 3.1.5). But the general principle of being a supervised learning method for doing score prediction that is conditional on characteristics of the input also perfectly applies to aggregate settings, where scores are measured over a collection of instances. We use the term aggregate assessors purely for notational purposes.

Aggregate assessors can be used for optimising the inference process –e.g. as in Jang et al. (2023), which is an anticipative model selector–, aggregate assessors are particularly prominent in the optimisation of the development process, i.e. helping in architecture search or AutoML, but also in e.g. deciding on a data-labelling plan under budget constraints (Srinivasan et al., 2022).

X. Song, Li, et al. (2024), Kokiopoulou et al. (2020), and Istrate et al. (2019) are all examples of aggregate assessors used for applications somewhere on the spectrum between architecture search, meta-learning, and AutoML.

4.2.5.3 Assessors

Generalisation	Granularity	Subject Info	Instance Info
Full*	Any	Any	Any

By now, we know what assessors are: instance-level score predictors trained in a supervised way on a collection of evaluation records, which condition on both subject and instance features.

There are, of course, more types of supervised machine learning methods that can be enumerated. We will therefore focus on specific demonstrations, and how they differ from each other.

For example C.-H. Lee et al. (2024) and Shnitzer et al. (2023) is a kNN-based assessor for instance-level model selection in an LLM context. It does not demonstrate subject generalisation, but does demonstrate the benefits of meta-learning in performance modelling.

Hu et al. (2024) also explores kNN-based assessors for LLM routing, but additionally compares a feed-forward neural network approach. Ong et al. (2024) on the other hand, is inspired by recommender-systems and uses matrix factorisation, but based on instance embeddings instead of ids to support instance generalisation, much like Two Tower recommender systems (P.-S. Huang et al., 2013).

Costa et al. (2023) and Drapal et al. (2024) are assessors in a tabular data setting, focusing on instance-level rejection, and Prudêncio et al. (2024) is also focused on tabular data, but uses assessors in combination with Shapley values and partial dependence plots to explain instance-hardness.

Lastly, V. Bhatt et al. (2022) is an assessor in the context of procedural generation of reinforcement learning tasks. In order to not waste time on training on overly difficult or easy environments, they employ an assessor model that estimates the score. The input passed to the assessor is a one-hot encoded representation of the image of the environment, which first predicts the expected behaviour of the RL agent, after which the behaviour plus representation of the environment is passed into a second stage, predicting the score.

4.2.5.4 Capability-Oriented Evaluation and Measurement Layouts

Generalisation	Granularity	Subject Info	Instance Info
Full[1]	Instance	Any	Features[2]

[1] Subject generalisation has not yet been demonstrated.

[2] Instances might need to be pre-processed by humans.

The idea behind capability-oriented evaluation is to measure task-independent capabilities that are robust across different conditions and difficulties. The various capabilities of interest are informed by domain knowledge, e.g. from the vast body of research in animal cognition, psychometrics and comparative cognition. Example capabilities include *object-permanence*, *memory*, or the *understanding of affordances* – the capability to identify what action-possibilities exist with particular object or set of objects, given an agent's specific physical properties and capacities.

This contrasts against previously discussed methods in the following way:

vs. task-oriented approaches: The conventional calculation of an aggregate metric on a hold-out set and most subject-generalisers are task-oriented in the sense that they aim to measure performance on a particular concrete task, not to make an assessment of the multiple capabilities that are required to solve such a task.

vs. anonymous latent factors: The item-generalisers or full-generalisers we have discussed that have an explicit notion of difficulty (or novelty) –including assessor models– do not measure explicit interpretable capabilities. Their representation of the subject's capabilities is either an abstract latent space of arbitrary dimension, just like factor analysis, or in a task-specific performance disaggregated over various difficulty levels.

vs. populational methods: Methods from psychometrics like IRT and factor analysis rely on populational performance to assess subject capabilities. Capability-oriented evaluation aims to assess the capabilities of a subject from their performance data alone.

If we measure subject capabilities, we can anticipate performance on new instances, given that we can derive the corresponding instance demands. Conversely, if we can relate subject design to their capabilities, we can anticipate performance for an arbitrary design. It should be noted that capability-oriented subject generalisation has not been demonstrated in practice.

Contributions. In Voudouris et al. (2023) we designed a digital environment for reinforcement learning agents specifically aimed at creating benchmarks capability oriented evaluation and comparative cognition. In Rutar et al. (2024) we created such suite of tasks, which was focused on measuring an agent's understanding of affordances, and in Voudouris et al. (2024) we investigated the object permanence of a concrete state-of- the-art reinforcement learning agent.

Burden et al. (2023) presents 'measurement layouts', a Bayesian framework for triangulation task demands and agent capabilities –also referred to as the cognitive profile– from specific observable instance meta-features, which are constructed based on domain knowledge of task demands and subject biases.[14,15] Measurement layouts are defined as 'directed acyclic graph that connects, through linking functions, the meta-features of a task characterisation with the cognitive profile of a system, in order to predict observed performance', and they have been implemented with Hierarchical Bayesian Networks (Murphy, 2023). A more detailed exposition of measurement layouts is out of scope, but we present fig. 4.7 as an example layout for illustration purposes.

Because of the dependency on domain knowledge, the provided generalisation is split between the Bayesian inference framework model and the human provided instance-meta features and capability dimensions. This makes for a good segue into our next section on human mental models.

4.2.5.5 Human Mental Models

Generalisation	Granularity	Subject Info	Instance Info
Full	Any	Features	Features

Human learning is very flexible with respect to the sort of evidence and information it can accommodate. We can interpret the scientific literature, do experiments, and connect various machine learning techniques purely based on the understanding from their inner working, rather than their empirical demonstrations. Humans have a rich model of the world, and that includes and affects the mental models of AI performance. We can anticipate the performance of new subjects, of new instances, and combinations of both.

[14]While features are observable, they are not necessarily easily derivable with automatic means, and might require human annotation.

[15]Subject features can also be used in theory, but that has not yet been demonstrated.

But there are of course also trade-offs and challenges, and they are four-fold: (i) human reasoning is not automatable, which invalidates certain high-speed or high-frequency applications, but also imposes a general resource cost; (ii) some problem spaces are hard for humans to reason in, e.g. high-dimensional tabular data or DNA sequences; (iii) there is a lot to learn about AI systems before we can accurately predict scores, which is a burden we can not place on lay people; (iv) it is not entirely clear if we can actually accurately predict scores, to the level we can rely on it.

For the latter, we last point, we refer to Chapter 6, but also Carlini (2023) is an interesting popular science experiment investigating how humans predict language model scores, while Meding et al. (2021, Sec. 3.4) is a short, but more formal investigation.

4.2.6 Miscellaneous

Not all methods neatly fit the organisation, but nonetheless fit the framework. Or they fit perfectly, but are very niche or specific, and thus not worth and entire section. Others do not fit the framework, but nonetheless predict scores in some way. All these will be briefly discussed here.

- Udandarao et al. (2024) find that the frequency of a certain concept in the training set determines (on a log-axis) the performance of a multi-modal model on an instance relating to that concept. Using a mechanism for mapping instances to concepts, and a mechanism for estimating the frequency of the concept in the training data, they can estimate scores. This is a full generalisation method using the training data of the subject as subject feature.

- The Hurst statistic of LLMs is predictive of their downstream performance (Alabdulmohsin et al., 2024). This can act as a subject generaliser.

- Ruan et al. (2024) uses a method similar to factor analysis for analysing scaling behaviour over several subjects and benchmarks in the NLP domain. They extract 3 factors, and find that the factors correlate log-linearly with the compute resources spend on training the subject. They then use the measure of compute as the uni-dimensional trait for fitting logistic function 2 free variables for each benchmark, much like in IRT. While expounded in the paper, scores need to be collected for the specific trained subject on other benchmarks before performance can be predicted for the subject. The target of prediction is performance on downstream capabilities, for which factor loadings are already computed. Unsurprising given the basis in factor

analysis, this is method an inner-generaliser, which is in contrast with other scaling analysis methods.

- Ye et al. (2023) is a collection of experiments for analysing scaling behaviour, requiring only inner generalisation (as the paper above). They explore matrix factorisation (inner generalisation), assessors (full generalisation), and a variety of methods in between, finding assessors to be the most accurate.
- An ELO rating as in chess, or any ranking score, e.g. as in ChatBotArena (Chiang et al., 2024) is a relative score predictor indicating which subject is preferable for a given instance. But there is no generalisation, as the predictions are equal for all unseen instances (and an unseen subject has no ranking).
- ConfidNet (Corbiere et al., 2021; Corbière et al., 2019) is a semi-anticipative method, where the image classification backbone is shared between score predictor and subject. This provides instance generalisation.
- Demonstrated in the image classification domain, Kaplun et al. (2022) is a method for constructing 'learning profiles', a mapping from global subject accuracy to both score and behaviour for a specific instance. This reminds us of item characteristic curves from IRT, where the global subject accuracy now takes the place of trait variable, and is no longer latent. The analysis done is similar to the one in Martínez-Plumed et al. (2019).
- InstaNAS (Cheng et al., 2020) is wonderful blend between many methods and applications we have already seen. It is a Neural Architecture Search procedure that produces a collection of subjects rather than a single one over which they then do instance-level model routing. The fact that the router is trained jointly with the subjects reminds of Mixture of Expert models, except that here the subjects can have wildly different architectures, which is not common for MoE subjects.

There also some methods that we have discussed before in other sections or chapters, and that do not need repeating, but we add here again for completeness.

- We discussed router models in Mixture of Experts already in section 4.1.3.3. These are instance generalisers.
- Datamodels have been discussed in the context of removing contamination (section 2.6.3). These are subject generalisers that observe as subject description a binary vector where each entry indicates whether the corresponding instance is present in the training set of the subject.

There are also some ideas that are worth mentioning, but for which a proper analysis is out of scope.

- As mentioned in section 4.1.3.4, there are methods for performance prediction, referred to as 'shielding', which are tailored to sequential decision problems. These tend to estimate the probability of a certain action in a certain state leading to a safety violation within certain number of steps. Bastani (2021), Könighofer et al. (2023), and Waga et al. (2022) are example methods.

- There is a large suite of research that does score prediction by attesting provable guarantees of performance or formally verified learning techniques. These techniques can for example make the claim that whenever an instance falls within a certain region of the input space, it is guaranteed to be classified correctly. Some example papers are Arora et al. (2013) and Celis et al. (2019); or X. Huang et al. (2017). The research area around robustness is closely related (Carlini & Wagner, 2017).

- There is a large research effort into non-anticipative score predictors for tasks with complex output spaces, e.g. natural language or images. For example LLM-as-a-judge techniques use LLMs to grade the responses of other LLMs automatically in the context of evaluation (Zheng et al., 2023) and similarly automatic 'verification' methods for during inference (Cobbe et al., 2021), or reward models for doing so during training (Lambert et al., 2024).

- The notion of trust is very related to performance prediction. Humans also trust subjects more if there is uncertainty –i.e. score estimation– attached to their predictions (U. Bhatt et al., 2021). But the concept extends further. We also build trust through institutional processes. If we know certain companies regularly get audits, report on their evaluation efforts, their commitment to fairness etc., we are more likely to trust the subjects they release, estimating their general performance to be higher, e.g. with respect to bias, even without having observed performance.

- The Matilda project is a tenacious multi-project effort of analysing the relation between problem characteristics and algorithm performance, applying both to optimisation problems like the travelling salesman problem and to machine learning. Muñoz et al. (2018) is an example of the latter. *Note that what the Matilda project refers to as an instance would be considered a dataset, task, or distribution in traditional machine learning nomenclature, as well as in our framework.*

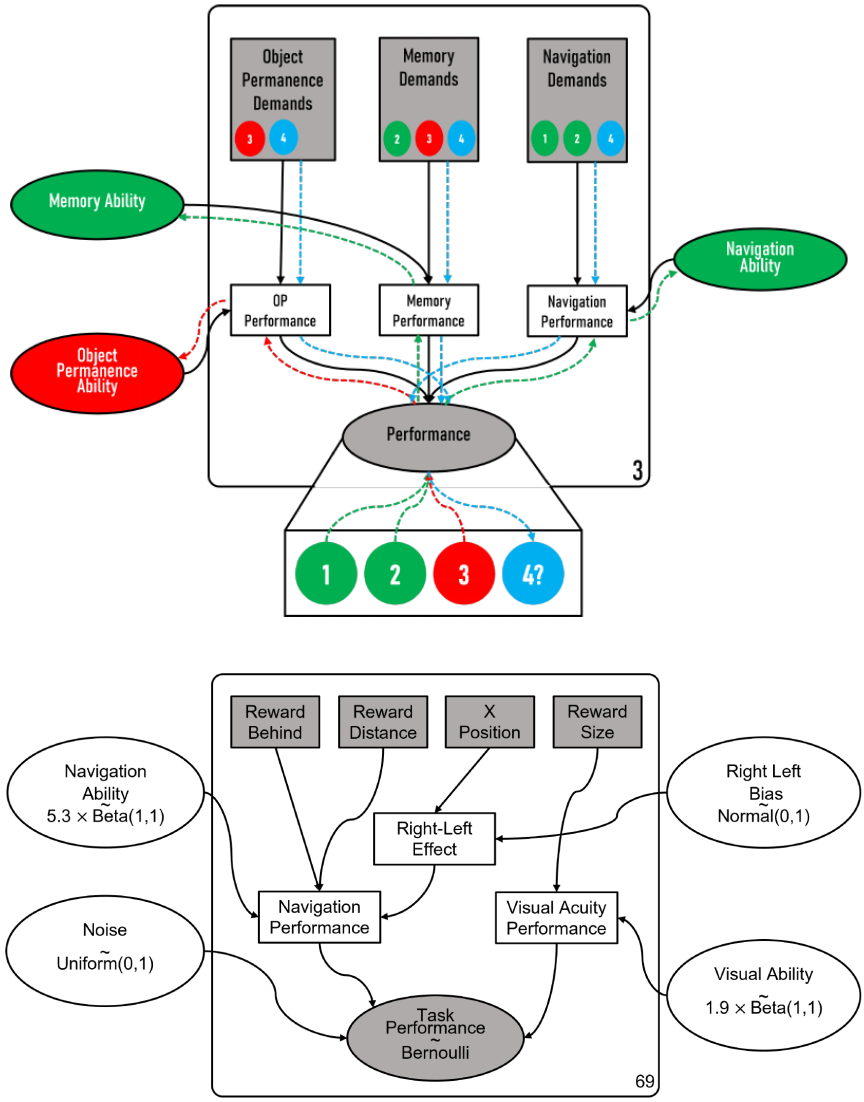


Figure 4.7: Example measurement layout. Both panels from Burden et al. (2023). (left) Illustrative depiction of measurement layouts. Bottom-up inference from three tasks (in green and red for success and failure respectively) leading to the cognitive profile. Top-down inference (in blue) predicting the outcome for the fourth task. (right) Concrete measurement layout as used in Burden et al. (2023).

4.3 Method Properties

Now that we have given a diverse exposition of score predicting systems, we can discuss the various properties that differentiate and connect them. There are a few major boundaries and features that differentiate score predictors.

We have already briefly discussed the granularity of the prediction, what information about the subject is used, and what information about the instance is used. We have also already discussed the generalisation types of the predictor –no, inner, subject, item, full–, but we will revisit these in more detail and explain how they can be derived from the levels of subject and instance info (section 4.3.1).

Afterwards, we discuss the levels of info about subject-instance interaction and how that relates to different kinds of anticipation (section 4.3.2). Lastly, the choice of prediction target also has various interesting implications and properties, which will be discussed in section 4.3.3.

4.3.1 Generalisation

The five types of generalisation used to organise the method section are visually depicted in fig. 4.8, and as you might have noticed in that method section, there was a pattern. The type of information that is observed about subject and instance determines how score predictors can generalise to new subjects, new items, and combinations of both. An identifier is devoid of meaning: there is no notion of similarity, and no room for inductive bias or generalisation. When observing a new id, nothing is known about it. Whereas when observing, say, a new RGB image that is only 20% darker but otherwise the same as an image already seen, what we have already learned about one image is likely to transfer to the other.

This means that when we observe only id's for a particular dimension, we can not generalise across that dimension, i.e. when only observing subject id's we can not predict scores for unseen subjects, and the same goes for instance id's and instance generalisation. Considering this, let us revisit the five types of generalisation to discuss this in more detail.

No generalisation: *The score predicting method does not make predictions for unseen (subject, item) combinations.*

The corresponding methods, aggregate performance metrics (section 4.2.1.1) and populational difficulty metrics (section 4.2.1.2) respectively only observe (subject id, task id) and (population id, item id), which explains their lack of generalisation.

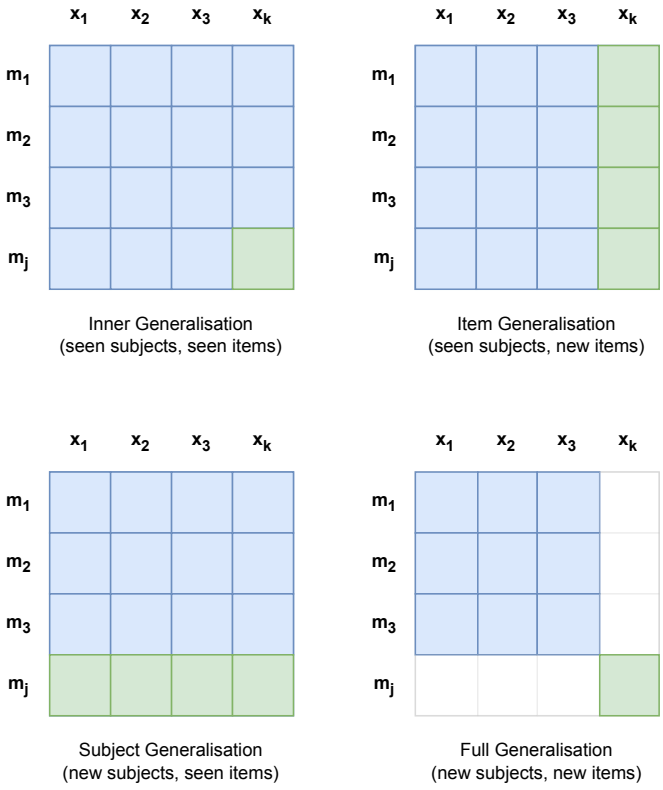


Figure 4.8: Different kinds of generalisation for score predictors. The table represents a set of evaluation records for a set of subjects $m_1...m_j$ (the rows) and a set of items $x_1...x_k$ (the columns). A cell is coloured green if the score predictor can make a non-trivial performance prediction for the pair (m, x), when having observed the scores associated with the blue coloured cells. Sparsity in the blue coloured cells is allowed. In the full generalisation case, the white squares can also be predicted, but the green corner is the highlight. Score predictors that can not generalise at all are not displayed.

There is still a predictive aspect though. Scores can be stochastic for a given `(subject, item)` pair (see also section 4.3.3.1), and the aggregate scores we measure are only a finite-data estimation of an usually infinite-data notion of performance. The accuracy of a *no generalisation* performance claim improves with more evaluation data.

Inner Generalisation: *The score predicting method can make predictions for an unseen `(subject, item)` combination only when it has observed scores for both the subject on other items, and scores for the item from other subjects.*

Inner generalisers like Item Response Theory (section 4.2.2.2) and matrix factorisation (section 4.2.2.1), like methods without generalisation, also only observe identifiers. So what is the difference? All inner generalisers are based on the assumption that there are one or more latent factors that can observe the explained performance. Scores are not predicted based on the similarities in the instance or subject, scores are predicted based on the similarity in their corresponding scores. If a subject consistently does well, it is expected to do well for other instances. But this assumption might fail, e.g. if another instance would be loaded on another factor, and this assumption can only be verified if we have observed the scores for other subjects on that instance.

Item Generalisation: *The score predicting method can make predictions for an unseen `(subject, item)` combination when it has observed scores for the subject on other items, but it is not necessary to have observed scores for this particular item.*

Item generalisers are based on the assumption that there are *observable* rather than *latent* instance features that have an effect on subject performance, or at least that if there is a confounding latent factor, that information about it can be deduced purely from the observable features. For this, a score predictor must of course observe these features. Intrinsic difficulty functions (section 4.2.4.2) are an example method.

Subject Generalisation: *The score predicting method can make predictions for an unseen `(subject, item)` combination when it has observed scores for the item from other subjects, but not necessarily from this particular subject.*

Symmetrically to item generalisers, subject generalisers assume there are observable features of the subject that correlate with or cause performance. Scaling laws (section 4.2.3.1) are an example method.

Full Generalisation: *The score predicting method can make predictions for an unseen* `(subject, item)` *pair while scores from neither the subject nor the item have been observed. Both item and subject features must be observed to allow this generalisation.*

Assessors (section 4.2.5.3) are an example method. Note that the full generalisation methods can usually also observe IDs instead of features, but that will affect their generalisation.

Either Generalisation: In addition to the above, a generalisation type could be possible where the score prediction method can make predictions for an unseen `(subject, item)` combination when it has observed scores for *either* the subject on other items, or for the item from other subjects. But we have not found any methods that work as such.

This notion of generalisation is similar to the problem of cold starts in recommender systems (Panda & Ray, 2022). As already discussed briefly in section 4.2.2.1, recommender systems predict whether a certain user will like a certain product for purchase or consumption, and are widely used in e-commerce webshops and media services. A recommender system estimates a score for each (product, user) pair, completing a matrix much like fig. 4.8, which is usually very sparse in the case of recommender systems, as the universe of products and instances is big.

The cold start term is used to refer to the challenges present when making a rating estimation for unseen products or users, which can be difficult as there is no observed history. That a cold start is an issue makes sense, as a popular suite of techniques called *collaborative filtering*, which encompasses matrix factorisation, is based only on user and item id's, which can therefore only perform inner generalisation (Ekstrand et al., 2011)! Unsurprisingly, other methods –under the name of *content-based filtering*– were developed to tackle cold starts, which do look at product features (e.g. tags), user features (e.g. demographic), or both (P.-S. Huang et al., 2013; Lops et al., 2011). The symmetries between score prediction and recommender system are plenty, and could lead to fruitful cross-fertilisation between the two.

4.3.2 Anticipation

In doing score prediction, the types of information observed about subject and instance can also be complemented by information about their interaction, i.e. the behaviour of the subject w.r.t. the instance. There are several degrees:

No information: *No subject-instance interaction is observed.*

The algorithm learns from the standard the standard $\langle \mathbf{m}, \mathbf{x}, r \rangle$ tuples, and can predict scores given only $\mathbf{m}$ and $\mathbf{x}$.

Partial behaviour: *The score predictor observes subject-instance interaction only in part, and not to the degree that subject behaviour can be graded.*

This could be for example a partial computation, where the score predictor observes model internal calculations, e.g. the activation of specific neurons in a neural network, but not its final output, as in ConfidNet (Corbiere et al., 2021), or for a sequential decision problem, where the learning algorithm observes part of the input/output trajectory, but does not require the episode to be completed.

Full behaviour: *The score predictor requires observing the full behaviour (e.g. classifier outputs) for the respective instances.*

Observing subject features so detailed as to allow simulating the subject –e.g. the coefficients of a regression– counts as observing full behaviour.

The amount of subject-instance interaction a score predicting method observes significantly impacts the applications it can be used for, and what the score predictor must learn in the first place.

When no subject-instance interaction is required before making a prediction, then a score predictor is **fully anticipative**. Being anticipative allows predicting scores without causing any effects induced by the subject behaviour, e.g. compute cost or time, but also more physical effects. For instance, when picking an optimal robotaxi, you only need to be driven to your destination once, it does not make sense to fully observe the behaviour of all taxis before making your decision. Presuming subject generalisation, a score predictor being fully anticipative also allows predicting scores for hypothetical subjects, which are not yet trained, but of which the design can be described; an essential property for e.g. scaling analysis (section 4.1.1.2). To make accurate predictions, a score predictor must learn to determine what features make an instance hard and a subject capable.

On the other hand, when the full subject-instance behaviour is observed, e.g. the final output of a classifier or the episode end of a sequential decision problem, the score prediction is no longer about anticipating quality, but is instead about (automatically) determining quality, i.e. grading, verifying, or judging. See for example Zheng et al. (2023). For a given instance, all information relevant to the determination of a score is captured in the behaviour. **The subject that generated the behaviour is irrelevant to that assessment, so the score predictor must learn nothing about it.**

A score predictor can also be **partially anticipative**, when subject-instance interaction is observed, but not "to completion". This happens for example when subject internal computations are observed, but not its final output, or as in sequential decision problems, when certain time steps have passed, but the episode or task has not been brought to an end. While only full anticipation allows for score prediction for hypothetical subjects or instances, the level of anticipation is thus on a spectrum, where less anticipation brings us closer to grading, and more anticipation strengthens the predictive aspect that requires us to learn about instance-subject interaction, as opposed to observing it. An alternative perspective is that the more behaviour we observe, the smaller the actual subject under evaluation is. If we observe the intermediate representation of a neural network, but not its classification head, what we are evaluating and predicting scores for is the classification head.

Anticipative prediction is the delimiter of our predictive viewpoint of evaluation, and we further expound on the motivation in the FAQ (appendix B.2). Figure 4.9 provides an overview. The notion of anticipation is also related to the difference between separated, dependent and integrated rejectors in machine learning with a reject option Hendrickx et al. (2024, Sec. 4-6).

Lastly, there is an elephant in the room. For the typical evaluation use case –testing a model to estimate its deployment performance–, the instances which we are predicting for are unknown. We do not yet know what the deployment instances look like. We might assume they follow a certain distribution, allowing us to sample from that distribution, but that assumption is a prediction. And it could be that we know some rough characteristics, e.g. that the task will involve some sort of reasoning, or that we expect to see more female patients, or more elderly ones. Until we observe the actual instances those expectations are prediction, not certainty, in the sense that a distribution shift occurring causes our test set –the chosen representation of the future– to no longer be representative. We are confronted with the fact that a significant portion of the prediction of scores, is to first predict what the data will look like.

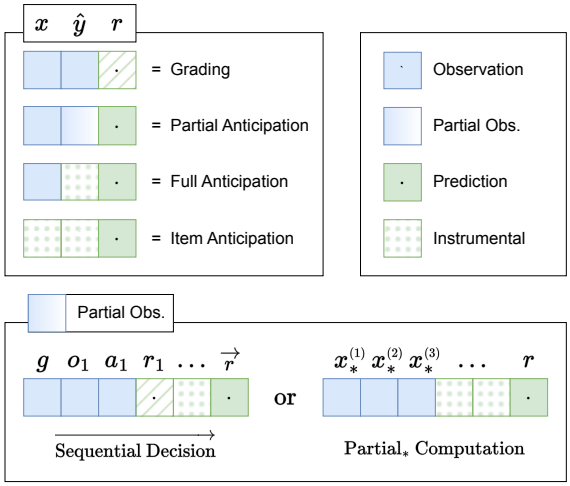


Figure 4.9: Degrees of anticipation. As usual x represents an item, while $\hat{y}$ represents a subject's behaviour for that item and r the corresponding score. The gradient blue box represents partial observation, i.e. where subject-instance interaction is observed, but only partly, as can for example happen in sequential decision problems and partial computation. The boxes with a dotted green pattern represent unobserved information that might be worth predicting as an instrumental goal, as it could be relevant to the determination r. Regarding symbols, g represents an (optional) observation of a goal or task description, while o_1, a_1, and r_1 represent respectively observation, action, and reward at first time step. The return –or any other final score– is represented by $\overrightarrow{r}$. Regarding partial computation, $x_*^{(i)}$ symbolises the subject's state after being processed by computational sub-graph i, e.g. the first layer of a neural network. We assume that the true label or optimal behaviour y is unknown.

For this reason, score predictors which use high-level representations of the task, e.g. capability-oriented evaluation (section 4.2.5.4), hold much potential.

4.3.3 Prediction Target

So far we have been talking about predicting scores in an abstract sense, and many things can conceptually be considered a score. Cross-entropy, the binary correctness of a classification, a reinforcement learning reward, or even the toxicity of a language model output all fit the bill. The formalisms we have introduced so far had to fit this abstract notion of score in order to make our analyses applicable to a wide range of domains and use cases.

But the metrics and scores we work with are determined by the domain of application and the methods under evaluation. In each setting (e.g. image classification)

and application there are various options that come with different trade-offs, and many of these trade-offs overarch different domains and use cases. An example is whether a score is absolute, e.g. chance of being correct, versus relative, e.g. a chance of being better some other bigger model; in instance-level model selection (section 4.1.3.2) ranking can be sufficient, as we can always pick the subject most likely to succeed, but absolute scores allow for better cost control, allowing us to take a quantified performance hit if the cost improvement is worth it.

In terms of overarching properties, we have identified the following:

Information — A loss like cross-entropy caries more information than binary correctness for example. The amount of information correspondingly affects the smoothness of the loss landscape of the score predictor. The assessor-like Nanny models (NannyML, 2024) for example predict losses. Schaeffer et al. (2023) and Schaeffer et al. (2024) provide convincing arguments for how scaling behaviour is more predictable when measuring scores that carry more information.

Continuity — Scores can be either continuous, like cross-entropy loss, or discrete, like binary correctness. Regardless of the information they carry, this also impacts which sort of models and methods are applicable for building score predictors, i.e. regressors or classifiers.

Proportionality — A relative improvement in the quality of the subject behaviour can lead to a relative improvement of the score (e.g. for cross-entropy loss), but this is not the case for all metrics. For example, consider binary correctness in a two-class classification setting, with an $\arg\max$ decision rule. A minor change in prediction quality, e.g. moving from 0.47 to 0.51 probability assigned to the correct class, changes the score significantly, from 0 to 1, while improving to a probability of 0.49 does not change the score at all. With 'proportional' we refer not to mathematical definition, but to the score being a strictly non-monotonic function of the quality of the behaviour.

Directionality — While most scores only reflect a magnitude, scores can have a direction as well, especially when they are based on errors (i.e. difference between prediction and gold label). An example is the absolute error $|1 - p(y)|$ versus the signed error $1 - p(y)$, the latter of which tells us in which direction the correct answer lies. Predicting a directional error allows for correcting the subject prediction, but predicting the sign can be difficult. To see why, we can consider fig. 3.3 again, where we encountered a subject doing regression under heteroscedastic noise $\mathcal{N}(0, x)$ as a function of instance x.

The sign of the error is impossible to predict, but the magnitude of the error follows the folded normal distribution $|\mathcal{N}(0, x)|$, for which we can estimate the mean. The direction and magnitude of the error can be split as separate targets, such that the difficulty of predicting the direction does not affect prediction of the magnitude.

Availability — A surprisingly relevant property is whether the score can be calculated from available data. While model developers or researchers sometimes release valuable evaluation data online, it is not guaranteed to contain the information you need. For example, both HELM (Liang et al., 2022) and BIG-Bench (Srivastava et al., 2022), two big NLP benchmark suites, published an extensive collection of evaluation records. While they included scores for their metric of choice, if you would want to learn to predict losses from this data, you are out of luck, as these records only recorded the subjects confidence in their output, not the confidence in the correct answer, and definitely not the confidence in all possible answers. As such, a loss can not be calculated.

Decision-based — Given a probabilistic subject $m(y|x)$, is the score based on a concrete decision $\hat{y}$, e.g. $\hat{y} = \arg\max m(y|x)$, or is it based on the probability distribution over y? A decision-based score, e.g. binary correctness, has lower information, but if the decision rule used in measurement of the score coincides with the decision rule we plan to use, we have all the information we need. On the other hand, if we predict log loss, we do know something about the expected confidence assigned to the correct class, but we do not know how that probability relates to the probabilities assigned to other classes. Consider the probability vectors $[0.3, 0.2, 0.2, 0.2]$ versus $[0.45, 0.55, 0.0, 0.0]$, with the true label being y_1. The loss in the first vector is higher, but the correct decision will be taken, while the loss in the second is lower, but a wrong decision will be taken. Schaeffer et al. (2024) provides interesting analysis on why predicting decision-based scores can be harder.

Absoluteness — Scores can be absolute in different ways, in that they contrast different notions of relative scores: (i) scores can be relative scores like the relative error $(y-\hat{y})/y$, which divides the error by the magnitude of the target, see also Park and Stefanski (1998); (ii) scores can be relative in that they are normalised by the error of a reference model; and (iii) scores can be relative in that they are a ranking of how one subject's output compares against another. Rank scores can be a viable target for score prediction use cases, for example for model selection (section 4.1.3.2,) or architecture search, e.g. Y. Xu et al. (2021).

Normalised/Standardised — After scores are assigned, they can additionally be normalised to fit a given range, or standardised to fit a a standard normal distribution. Especially when learning from evaluation data across multiple tasks, multiple metrics might be in use, each with its own mean, range, and variance, etc. Normalisation of the scores across the different tasks can help facilitate score prediction, as is done for example in OmniPred (X. Song, Li, et al., 2024).

Bounded — Relatedly, if a score is unbounded on some side, e.g. some RL rewards have a range $[0, \infty)$, the score predictor potentially needs to extrapolate to entirely unseen scores.

Plurality of Subject Does the score predictor predict for a single subject, or instead for multiple subjects simultaneously in a vector of scores? Multi-subject predictions can be useful to save on inference costs and time, and for example in Mixture-of-Expert models, the router layer typically assigns confidences/weights to all experts in a single forward pass (section 4.1.3.3).

Granularity — Simply, is the prediction target at the instance level, or is it an aggregate metric? This affects how much of the variation in scores across items can be explained. Some aggregate metrics, e.g. AUROC, do not have an instance-level equivalent.

Stochasticity — For a given instance and subject, is the score stochastic or not. See section 4.3.3.1 for discussion.

Apart from all the scores and metrics that are commonly used to report on subject predictive power (log loss, accuracy, squared error, ...), some other interesting prediction targets have been proposed. In the context of classification, Corbiere et al. (2021) and Corbière et al. (2019) use the True Class Confidence, meaning that the prediction target for the score predictor is the confidence the subject will assign to the true class, and Ilyas et al. (2022) use the Correct-class Margin: (logit for correct class) − (highest incorrect logit), selected because for their data it followed a Gaussian distribution, which they reasoned would align with their ordinary least squares score predictors.

Joined in their application to scenarios with exactly two subjects at hand, Kag et al. (2022) uses $m_a(x) \neq m_b(x) = y$ for deferral from subject m_a to subject m_b, but only if both m_a is expected to be wrong and m_b is expected to be right, Ding et al. (2023) uses the quality difference between the two subjects as a target, and Ong et al. (2024) predicts the chance that the response of m_a will be preferred over that of m_b by human raters.

4.3.3.1 Sources of Stochasticity

There is good reason to consider modelling scores to be variable for a given instance since there are potential sources of randomness in every element of their measurement. Firstly, instances can be stochastic, in that multiple situations can map to the same instance representation $\mathbf{x}$ if our representation cannot capture the full complexity of the problem, as would for example be the case for most systems operating in the physical rather than the digital world.

The same holds for subject representation $\mathbf{m}$. While one can always add a unique identifier to refer to a particular (trained) subject, for some use cases this is not applicable. For example in Bayesian optimisation section 4.2.3.3, no subject id is used and subjects with different performance will be produced from the same hyper-parameter based representation, leading to a distribution of scores for a given $\mathbf{m}$. Subject systems can also be stochastic in nature, e.g. generative language models, which use a temperature parameter to control randomness of the output. Ding et al. (2023) specifically compares stochastic and deterministic modelling of the scores, finding stochastic modelling to consistently outperform (albeit not by much), when applied to LLMs to instance-level model routing.

Lastly, the scores itself can be considered stochastic in some settings, in that for example different users have different preferences or might disagree. Although arguably this can be considered as under-specification of the instance, by not taking the user into account as a feature, which is something that section 3.4 has talked about already.

In general, the stochastic nature of a score, is closely related to the notion of 'repeatability' or 'test-retest reliability' in measurement theory, and more broadly, to the uncertainty associated with the measurement (JCGM, 2008). So while a score predictor can consider both probabilistic and point estimates, there are good reasons for considering probabilistic outputs other than wanting to represent score predictor confidence.

4.4 Discussion

We have really connected a vast body of literature into a single perspective. There is now a connection between model selection, curriculum learning, routing, scaling laws, meta-learning, mixture of experts, distribution shift estimators, recommender systems, item response theory, and many more ideas that have previously been considered as separate topics. If we know what sort of information about the subject and instance is used, we can quickly deduct the corresponding

generalisation type, and these types have proven very useful for grouping related methods, e.g. factor analysis and matrix factorisation, or quite the opposite, disentangling them, e.g. populational vs. intrinsic difficulty measures.

A lot of the applications we highlighted here relate to optimising resources, be it for development or for inference. For optimising development, the cost savings are in that order that they allow reaching otherwise unachievable levels of performance, since our budget is usually finite. For optimising performance of single-step problems, e.g. image classification, the motivation is always purely for resource-related purposes, as with regards to safety, self-assessment or post-hoc verifiers could do the trick.

But for sequential decision problems and batch deployment decisions, this is not the case, and we might wonder if our motivation of anticipative performance prediction holds, as we have not been able to reference a rich literature. However, we believe this to be case not because there has been little need, but rather because there have been too few solutions. Why we do we not see autonomous agents acting on our behalf in our digital environments (Xie et al., 2024)? Their performance has already reached impressive levels. But if we predict performance in aggregate, measured performance needs to be near perfect in order for deployment to be safe, and even then, performance is susceptible to distribution shift, breaking that safety estimation. Both perfect performance and perfect robustness to distribution shift are unlikely to be solved soon, implying that, to deploy these agents safely, we need anticipative and granular performance prediction instead. Barring shielding techniques (Waga et al., 2022), as far as we have found, sufficiently accurate score predictors do not exist yet, especially not for general-purpose agents.

We should thus see the lack of referenced literature not as a detriment to the applicability of anticipative score predictors, but instead, as extra motivation.

Chapter 5

Language Model Score Predictors

Assessor models (sections 3.1.5 and 4.2.5.3) have popped up naturally as result of our framework. They have recently come into use through various efforts, both influenced by our work (Costa et al., 2023; Y. Zhao et al., 2024) and independently (Hu et al., 2024; NannyML, 2024; Shnitzer et al., 2023). But there is a vast design space: prediction target, data sources, training regimes... Their overall utility is not well established yet, and not much is known about the general factors that influence their performance. As a basic machine learning method, there are some obvious intuitions to be borrowed from the literature: e.g. use more data, scale up assessor size and out-of-distribution capabilities are difficult to achieve. As a second level predictive model, there are some additional more unique points of interest, e.g. does populational data help or harm predictive power? Does score predicting ability for a certain task correlate with the ability of the architecture to learn the task as subject? Do the *shots* –examples given to the subject– distract the assessor or make score prediction more difficult in general?,

We have focused on language models (for an introduction, see section 2.1), both as subject systems and as score predictors, due to their general relevance and the various challenges associated with their evaluation. Our experimental setting fine-tunes DeBERTa models (P. He, Gao, & Chen, 2021) as assessors on the evaluation

results of generative language subject models with up to 128 billion parameters. Our analysis spans more than 100 tasks from BIG-bench (Srivastava et al., 2022). We contextualise the (anticipative) assessor's performance by comparing to the (non-anticipative) subject's self-assessment, i.e. the confidence assigned to their outputs. Figure 5.1 provides an overview of this score predicting task. The figure presents four instances where an assessor model ($a(r|m, x)$) accurately predicts the expected performance r (success or failure) of the language model ($m(\hat{y}|x)$) for the given input x.

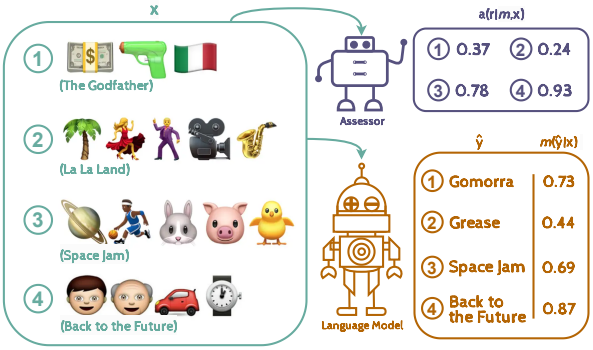


Figure 5.1: Illustrative example of the score prediction problem. The four instances are from the emoji_movie task (Srivastava et al., 2022), where the goal is to guess the title of popular movies from their plot descriptions written with emojis. The assessor $a(r|m, x)$ correctly anticipates the language models $m(\hat{y}|x)$ failures in cases ① and ② and its success in ③ and ④.

5.1 Research Questions and Findings

The key questions are address in this chapter are:

Q1 Can we learn to predict the performance of language models at the instance level? *Yes.*

Q2 Can we similarly learn to predict performance while not having seen evaluation data for the respective task, i.e. out of distribution? *Not sufficient for practical use, but generalisation is non-negligible.*

Q3 Is assessor performance correlated with the ability of its architecture to learn the original task? *Seemingly, but there is high variation among data setups.*

Q4 Does learning from the evaluation results of multiple tasks improve the assessor? *Yes, but it depends on the task.*

Q5 Does learning from the evaluation results of multiple subjects improve the assessor? *Yes, but it depends on the task.*

Q6 How does scale, of both subject and assessor, impact predictability? *Size of the subject has no impact on predictability, and at the scales we investigate, neither does the size of the assessor.*

Q7 Does the presence of few-shot examples in the input impact the assessor's performance? *Unclear.*

We find that the assessors can match and even exceed the subjects' confidence in both refinement and calibration (Q1), anticipating failures at near perfect levels for some tasks. Out-of-distribution performance is –not unexpectedly– lacklustre, although calibration is surprisingly good (Q2). There is correlation between an architecture's ability to do score prediction and its ability to solve the original task, but high variation in results might make the result untrustworthy (Q3). We also find that for performance prediction it can be beneficial to learn from the scores on multiple tasks or to learn from the scores of multiple subjects, but both depend on the task at hand (Q4,Q5). We find that large and small subject systems are equally predictable, showing promise for the scalability of the predictability problem, but in our setup larger assessor models are not necessarily better (Q6). We find no clear relation between the number of examples in the input and score predictability (Q7).

5.2 Experiment Setup

All experiments share the setup and data source described below.

5.2.1 Notation Recap

Let us define a supervised task, such as classifying the sentiment of movie reviews, as a set D of labelled instances $(x, y) \in \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ represents the input space (e.g., the movie review text) and $\mathcal{Y}$ represents the target space (e.g., the sentiment). A *subject system* (or subject) $m(\hat{y}|x)$ is a function $\mathcal{X} \to \mathbb{P}_{\mathcal{Y}}$. Here, m maps an input x to a probabilistic output over possible labels $\hat{y}$, with $\mathbb{P}_{\mathcal{Y}}$ referring to the space of probability distributions over $\mathcal{Y}$ (either discrete or continuous). Essentially, m is the model that performs the task (e.g., sentiment classifier).

Let M denote a population of such subject systems: $\{m_1, m_2, \dots\}$. We use the term *evaluation results* to refer to a set $\dot{R}_D^M$ of records (m, x, r), where each record consists of a subject system (m), an input instance (x) and the score m received for its output on x (r). For example, an evaluation score r could be a binary value $\mathbb{1}[\arg\max m(\hat{y}|x) = y]$, which indicates whether m correctly predicted the target y.

An assessor model, typically denoted $a(r|m, x)$, is a function $\mathcal{M} \times \mathcal{X} \to \mathbb{P}_{\mathbb{R}}$, where $\mathcal{M}$ represents features describing a subject system (e.g., number of parameters, unique identifier). We use a_D^M to indicate that the assessor has been trained with evaluation results for the population M on the task D. In practice, assessors often work with simplified versions of this general formulation. For instance, if M contains only a single model and all scores r are binary, the assessor's formulation can be simplified to: $\mathcal{X} \to \mathbb{P}_{\{0,1\}}$.

5.2.2 Data

All the evaluation results used for training assessors are sourced from BIG-bench (Srivastava et al., 2022). The BIG-bench repository provides the instance level outputs for a collection of text-to-text Google LMs across 200 collected natural language tasks. Depending on the task, results are present from the 0-shot to 3-shot setting. The collection spans various model sizes, two temperature settings, and two architectures: a decoder only dense transformer (BIG-G) based on the LaMDA architecture (Thoppilan et al., 2022) and a sparse Mixture-of-Experts model (BIG-G sparse) that is of similar design to (Zoph et al., 2022). More information on the BIG-G network architecture is available in (Srivastava et al., 2022). One model family is sufficient for our experiments, so we only use the evaluation results of the BIG-G dense model. See Table 5.1 for a summary of the models used. As in the BIG-bench paper, we only use the outputs that were sampled with temperature set to zero. We use consider 113 multiple choice tasks, as well as 48 generative tasks that use an exact string match to grade the answer, totalling 161 task. We remove all tasks from analysis for which the largest model performs poorly. The criterion is set as a minimum accuracy of 'random chance + 5', where random chance depends on the specific task, and equals 1 divided by the number of targets for multiple choice tasks, and 0 for generative tasks. For many of the tasks not meeting the criteria, we found that the BIG-G models have near constant outputs, e.g. always answering option A. From a score predicting perspective, this is not interesting, as it requires solving the original task, rather than identifying the weaknesses of the model. Assessors achieved AUROCs of 0.90+ on these tasks. While interesting to note assessors can identify these constant subjects, their inclusion would muddy the other research

questions. This filtering leaves 62 multiple choice tasks and 23 generative tasks, for a total of 85 tasks. In total, evaluation results are available for ~40K instances for each of the 12 BIG-G models, which range in size from 2 million parameters to 128 billion. The ~40K available records are not guaranteed to contain the same instances or the same amount of instances across 0-, 1-, 2-, and 3-shot settings, although the instances are the same across models. Everywhere except in section Q7 (Results) we only use 3-shot results. For a specific task, subject system, and shot setting, there are always 1000 or less instances available, with an average of 480.

Table 5.1: Overview of BIG-G models used in BIG-bench (Srivastava et al., 2022).

Model	#Parameters
BIG-G dense	2M
BIG-G dense	16M
BIG-G dense	53M
BIG-G dense	125M
BIG-G dense	244M
BIG-G dense	422M
BIG-G dense	1B
BIG-G dense	2B
BIG-G dense	4B
BIG-G dense	8B
BIG-G dense	27B
BIG-G dense	128B

5.2.3 Setup

The target scores that the assessor predict are binary: 1 if the subject model's answer is correct, and 0 if it is not. More formally, this can be expressed as $\mathbb{1}[\arg\max m(\hat{y}|x) = y]$, where $\arg\max m(\hat{y}|x)$ is the model's most likely prediction. Therefore, the assessor's task is a binary classification problem. The input to the assessor is exactly the same as the textual input to the subject system. If $|M| > 1$, it is potentially prepended with features that describe the subject for which it should predict scores (e.g., number of parameters).

For the assessor, we always use a pre-trained DeBERTa model (P. He, Liu, et al., 2021), specifically DeBERTaV3 (P. He, Gao, & Chen, 2021), which is then finetuned for the performance prediction task on the BIG-bench evaluation results. This decision was based on a number of considerations. DeBERTa models provided an standard, controlled and consistent means of exploring these questions (H. Wang et al., 2023). In addition, encoder-only models such as DeBERTa are generally more computationally efficient (K. Clark et al., 2019) than large

sequence-to-sequence models such as Llama-2, allowing us to conduct extensive experiments on a variety of tasks within a reasonable time frame and computational budget. Computational efficiency is also a requirement for various use cases, e.g. model routing (Hu et al., 2024). Furthermore, the use of BERT-based models is well studied and provides a strong baseline for performance prediction tasks (see, e.g., (A. Wang et al., 2019)), allowing us to ensure the comparability and traceability of our results.

The pre-trained DeBERTa model is provided by HuggingFace Model Hub[1] and later finetuned with the HuggingFace TRANSFORMERS library (Wolf et al., 2020) using the PyTorch backend. Unless otherwise specified, we use the base size with 86 million parameters, a batch size of 32 and default values for all hyperparameters, with the exception of the learning rate, which is set to 2×10^{-5}, and the warmup, which is set to 3000 steps. These hyperparameters were selected after initial exploratory experiments. We also conducted early exploration with similar models, including BERT (Devlin et al., 2019), ALBERT (Lan et al., 2020), RoBERTa (Y. Liu et al., 2019), and GPT-2 (Radford et al., 2019), but results were very similar, with all models achieving anywhere between 0.78 and 0.79 AUROC on the full dataset. The DeBERTa model can easily be swapped for any of the other models we have mentioned. A HuggingFace AutoTrain[2] run did not find any improved configurations. All experiments ran on a single NVIDIA GeForce RTX 3090 GPU with 24GB memory size.

Due to the limited amount of instances available for many tasks, we do a 80/20 train/test split without a separate validation set, although a different random split was used for the initial hyperparameter model and model selection. The split is created by holding out 20% of the instances of each task. This setup therefore measures generalisation across instances, as all tasks are present in both train and test set. We refer to this as an *instance-level split*. Only in Sec. 5.3, which details assessor generalisation across task boundaries, we do a *task-level split* instead where we hold 20% of tasks out entirely.

We have fine-tuned DeBERTa for at least 5 epochs, but selected the parameter values at the epoch with lowest test loss, which was always well before the fifth. The fact that test set was reused for this purpose should be taken into account when interpreting results.

[1] https://huggingface.co/models
[2] https://huggingface.co/autotrain

5.2.4 Baselines and Metrics

The class balance for the assessment problem depends on the accuracy of the subject system. For instance, if the subject has an accuracy of 85%, the class balance for the assessor's classification task is correspondingly 85% successes and 15% failures. Because accuracies vary wildly across tasks and subjects, we focus on the Area under the Receiver Operating Characteristic Curve (AUROC) metric, which measures the ability of the assessor to discriminate between successes and failures. AUROC is stable and not affected by class imbalance, making it a reliable metric for evaluating refinement. A higher AUROC indicates better discriminative performance, with 0.5 denoting random guessing and 1.0 signifying perfect discrimination. We additionally observe Brier scores (BS) (Brier, 1950), which measures the mean squared difference between predicted probabilities and the actual outcomes. BS combines refinement and calibration. Specifically we employ the formulation as proposed in Dimitriadis et al. (2021) and its decomposition into *miscalibration*, measuring the difference between predicted probabilities and the true probabilities (MCB, ↓ lower is better), *discrimination ability*, evaluating how well the predictions differentiate between instances that result in success and failures (DSC, ↑ higher is better), and *uncertainty*, which represents the inherent uncertainty in the prediction problem (UNC, neutral). The latter is a neutral term and does not directly impact the interpretation of the assessor's performance. We only report the *miscalibration* (MCB) to focus on calibration, as discrimination is already captured by AUROC.

To contextualise the assessor's performance, we compare it to the *self-assessment* abilities of the BIG-G models. Some natural baselines are provided by the subject's confidence, i.e., the probability the subject system assigns to its output. This confidence is often interpreted as an estimation of the probability of correctness, similar to the outputs of the assessor. The BIG-bench framework provides probabilities for multiple-choice questions in both absolute and normalised forms:

- Absolute probabilities: These refer to the probabilities assigned to each of the multiple-choice options. The sum of these probabilities does not necessarily equal 1, as the subject can assign non-zero probabilities to texts that are not valid options.
- Normalised probabilities: These are obtained by applying a *softmax* operation to the absolute probabilities, ensuring they form a proper probability distribution over the answer options.

We refer to these baselines as ABSOLUTE-SELF and NORMALISED-SELF respectively. We also create a third baseline, TASK-STATIC, which assigns a probability of success

equal to the subject's accuracy for the task to which the question belongs. For instance, when predicting the success probability for a set of questions from five different tasks, the TASK-STATIC baseline will output five distinct probabilities, one for each task. This baseline is perfectly calibrated but lacks refinement within a single task. However, it typically achieves a Brier refinement of ~0.02 and an AUROC of ~0.70 due to its ability to discriminate between tasks. Table 5.2 shows a summary of these baselines.

Table 5.2: Summary of the baselines used for comparing assessor's performance.

Baseline	Description
ABSOLUTE-SELF	Probabilities assigned to each multiple-choice option; may not sum to 1.
NORMALISED-SELF	Softmax-normalized probabilities forming a valid distribution over options.
TASK-STATIC	Predicted success probability equal to the subject's accuracy for the task.

5.3 Results

We follow the experimental questions sequentially. Following the Science paper's guidelines for AI evaluation reporting (Burnell, Schellaert, et al., 2023), all code, data, and instance-level results are available[3] for the sake of replicability/reproducibility. Aggregate performance metrics ('total') are always calculated over the instances of all BIG-bench tasks together. In other words, we do not average the metric across tasks, but consider all instances as part of a single set during calculation. This of course means that tasks with more instances available weigh more heavily in the aggregated result. When we present results disaggregated per task, only the tasks with more than 100 test instances are displayed.

Q1: In-distribution learning

In our first experiment, we focus on evaluating the largest member of the BIG-G model suite, the 128-billion-parameter variant denoted $M = \{m_{128b}\}$. This model represents our most powerful subject and provides a robust benchmark for comparison. To measure the generalisation of the model across tasks, we train an assessor on the results of a comprehensive set of more than 80 tasks. Figure 5.2 illustrates the assessor's performance by comparing the difference in the area under the receiver operating characteristic curve (AUROC) between the assessor and the NORMALISED-SELF baseline. Remarkably, the DeBERTa assessor demonstrates the ability to exceed the AUROC of BIG-G, which is particularly noteworthy given that it does so after fine-tuning on fewer than 800 samples per task. Although it should be noted that the BIG-G models were trained

[3] https://github.com/wschella/lass

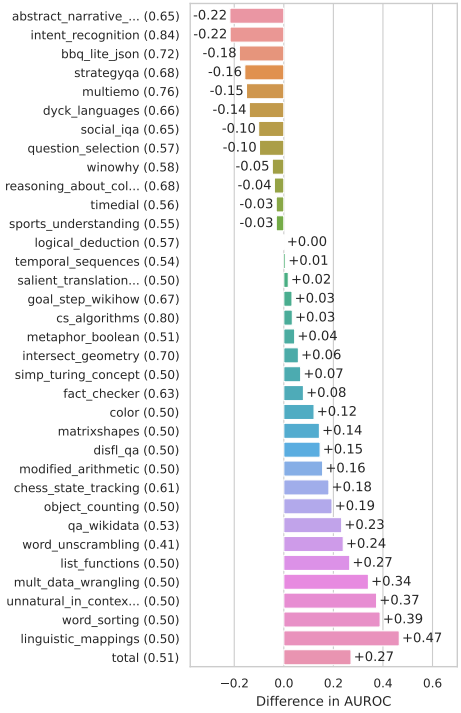


Figure 5.2: Assessor vs. BIG-G 128b generalisation. Difference in AUROC between the assessor and the NORMALISED-SELF baseline for BIG-G 128b. Between parentheses next to the task name is the NORMALISED-SELF baseline AUROC, while the bars represent how the assessor performs w.r.t. this baseline. (Q1)

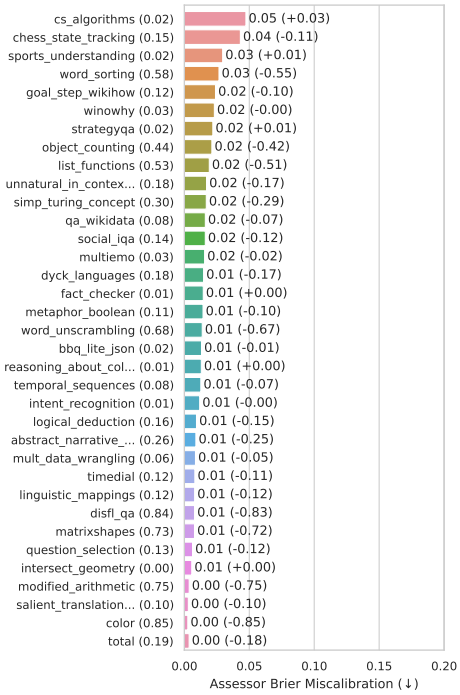


Figure 5.3: Assessor vs. BIG-G 128b miscalibration. The Brier score miscalibration error (MCB) of the assessor. Between parentheses next to the task name, we show the NORMALISED-SELF baseline for BIG-G 128b. Between parentheses next to bar label is the difference between assessor and NORMALISED-SELF MCB. (Q1)

unsupervised –without direct exposure to these specific tasks– their performance is often outperformed by the assessor.

The experiment reveals a predictive element in the performance of language models across tasks. Take the example of the 2-option task labelled LINGUISTIC_MAPPING; the BIG-G 128b model achieves a random 0.5 AUROC, while the assessor is close to perfection with 0.97 AUROC. This finding suggests that the assessor gains insight not only into the task but also into the nuanced ways in which BIG-G falters, mastering both the task and the error profile of the model.

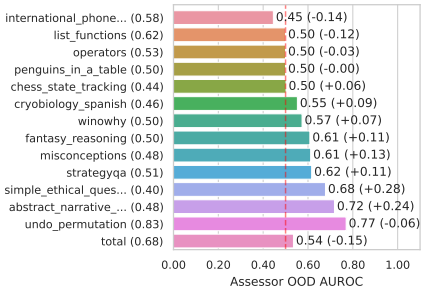


Figure 5.4: Assessor out-of-distribution generalisation. AUROCs of the assessor for a held out 20% of tasks. The single subject under assessment is BIG-G 128b, and its NORMALISED-SELF baseline is provided between parentheses. The dashed red line marks the AUROC for chance performance (0.5). (Q2)

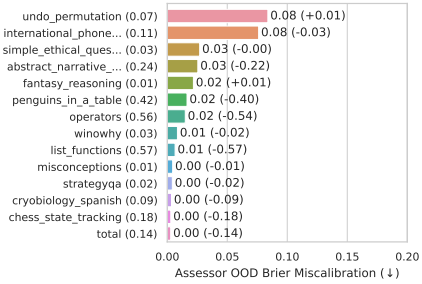


Figure 5.5: Assessor out-of-distribution miscalibration. The Brier score miscalibration error (MCB) of the assessor for out of distribution tasks. Task names are accompanied by the BIG-G 128b NORMALISED-SELF baseline. Between parentheses next to bar label is the difference between assessor and NORMALISED-SELF MCB. The rightmost column contrast BIG-G's accuracy with chance accuracy.

Building on the primary results, Figure 5.3 presents the Brier score miscalibration error (MCB) of the assessor. The MCB for the NORMALISED-SELF baseline is displayed in parentheses next to each task name, while the bars represent the difference in MCB between the assessor and the baseline. This figure reveals that the assessor achieves near-zero calibration errors across most tasks. The assessors not only improve on AUROC but also perform significantly better in calibration compared to the NORMALISED-SELF baseline. Interestingly, the TASK-STATIC baseline, which assigns task-level success probabilities based on subject accuracy, can achieve similar calibration but lacks the depth of refinement that the assessors provide.

Q2: Generalisation Across Tasks

Similar to the first experiment, we take the evaluation results of only the largest BIG-G model, i.e. $M = \{m_{128b}\}$. Unlike the previous experiment, we perform a task-level split, holding out 20% of the tasks entirely. This experiments demonstrates potential out-of-distribution generalisation, or rather, as can be seen in Fig. 5.4, the lack thereof. For most individual tasks, the assessor's performance improves on the baseline, showing above random performance and thus interesting levels of out-of-distribution generalisation. The aggregate AUROC is however lower than the NORMALISED-SELF baseline, and both are lower than the TASK-STATIC

baseline, which achieves a 0.68 AUROC solely by discriminating among tasks[4], suggesting out-of-distribution generalisation for both baseline and assessor is still lacklustre. This answers Q2 –can we learn to predict performance without seeing evaluation data for the respective task, i.e., across task boundaries?– with a nuanced negative, at least for measurements of refinement.

The reasons for this could be manifold. It may be that the model lacks the ability to extrapolate key indicators of success across different tasks, or it may be that the assessor fails to capture these indicators accurately. The features that indicate success or failure might not transfer across tasks, or perhaps the assessor does not pick up on them effectively.

If we turn our attention to the calibration aspect, as shown in Fig. 5.5. This figure presents the Brier score miscalibration error (MCB) for out-of-distribution tasks where the bars indicate the difference in MCB between the assessor and the NORMALISED-SELF baseline. Here, the assessor demonstrates commendable performance in terms of calibration, achieving near-zero calibration errors across seven out of eleven tasks, even for out-of-distribution data. This panorama shift suggests a dichotomy: although the assessor struggles with refined prediction, it exhibits impressive calibration capabilities.

Q3: Correlation with task performance

To deepen our understanding of why the predictability varied so much across tasks, we designed an experiment to compare two types of models: assessors (again for BIG-G 128b only) and a collection of new subjects that learn the original task. Both the assessors and the subjects were designed using the same DeBERTaV3 architecture (P. He, Gao, & Chen, 2021). DeBERTa as architecture was not designed for generative seq2seq tasks, so we restrict to multiple-choice tasks only. For each task, we trained an assessor to predict the performance of BIG-G 128b and a subject to learn the task itself. The training instances for the subject systems are the same ones as used for training the assessors, but now the target is the correct multiple choice option, instead of the score of BIG-G 128b.

To assess how well the subjects learned their respective tasks, we monitored their multi-class AUROC performance. Figure 5.6 shows the relationship between the AUROC scores of the DeBERTa subjects fine-tuned on each task (x-axis) and the AUROC scores of the assessors trained to predict performance for BIG-G 128b (y-axis). Each dot in the figure represents a different task. We plot both 0-shot

[4]This baseline is of course not available in many scenarios, as it requires labels to calculate the mean score. We simply highlights that both assessor and self-assessment do not reach even this very course prediction quality.

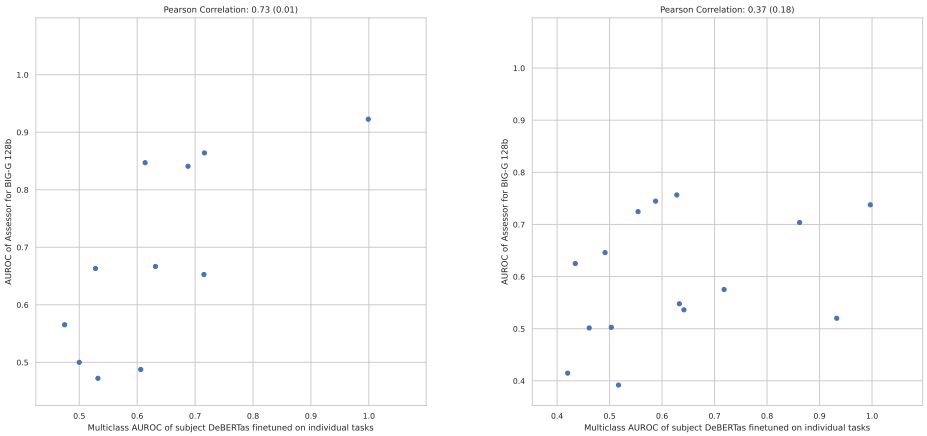


Figure 5.6: Correlation between subject and assessor performance. Left with 0-shot prompts. Right with 3-shot prompts. The correlation between, on the x-axis, the multi-class AUROC of DeBERTa subjects finetuned on each task individually, and on the y-axis, the AUROC of assessors trained on each task individually that predict performance for BIG-G 128b. Each dot represents a task. Note that the assessor does not predict performance for the subjects on the x-axis. (Q3)

and 3-shot results to have adequate data. A strong correlation would mean the predictive ability of assessors is influenced not only by the characteristics of the subject in question, but also by the characteristics of the task itself and the data representing it. But at correlation coefficients respectively 0.73 (p-value 0.01) and 0.37 (p-value 0.18), there is significant variation depending on the number of shots. While the results for 0-shot seem to be clear, for 3-shot prompts there is no significant conclusion. Combining the data points results in 0.5 correlation at a p-value of 0.01. We conclude that a correlation is likely, but a determination of the intensity and factors influencing the variation requires further research.

Q4: Using Results from Multiple Tasks

How does the performance of an assessor trained on all tasks compare to a collection of task-specific assessors? In this experiment, we explore the potential benefits of multitask learning compared to task-specific specialisation. Specifically, we investigate whether training an assessor on multiple tasks simultaneously offers any performance advantages over training separate assessors for each individual task, provided the use case allows for such flexibility. To achieve this, we train individual assessors using evaluation results from each task separately (as described in Q3) and compare their performance to that of the multitask assessor from Q1.

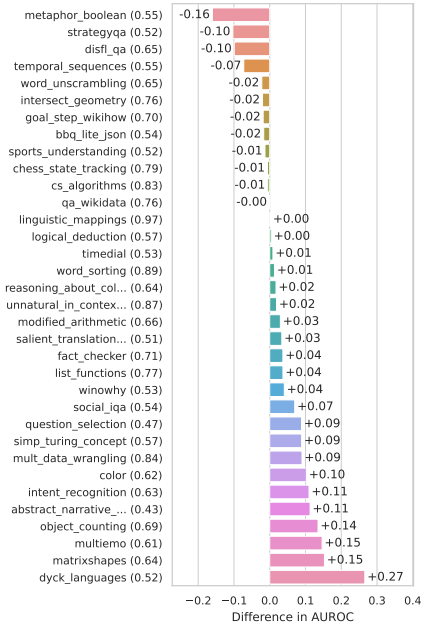


Figure 5.7: Multi-task vs. single-task assessors generalisation. Difference in AUROC between the multitask assessor from Q1 and a collection of task-specific assessors. Between parentheses next to the task is the multitask AUROC, while the bars represent how the individual task-specific assessors perform w.r.t. this baseline. (Q4)

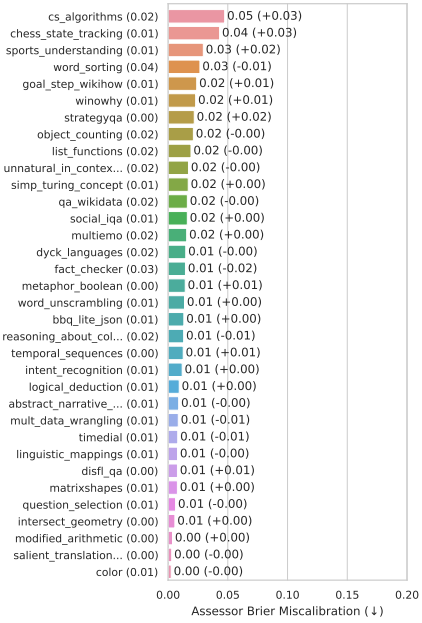


Figure 5.8: Multi-task vs. single-task assessors (miscalibration). The Brier score miscalibration error (MCB) of the single task assessors. Task names are accompanied by the multi-task assessor reference. Between parentheses next to bar label is the difference between single and multi-task MCB.

The multitask assessor is trained on the combined data from all tasks, while the task-specific assessors are trained only on data from their respective tasks. Apart from the training data, no other differences exist between the multitask assessor and the collection of task-specific ones.

Figure 5.7 illustrates the performance comparison between the multitask assessor and the task-specific assessors. The figure shows the difference in AUROC scores and it can be seen that task-specific assessors both regress and in improve in performance compared to multitask assessors, sometimes with serious differences. So while the results from Q2 indicate that there is little generalisation for out-of-distribution tasks, this experiment shows multitask learning can provide some benefit to in-distribution generalisation, although it seems to depend on the task.

The improved or regressed performance of the multitask assessor can be attributed to several key factors. Firstly, training on multiple tasks could provide a form of regularisation that helps prevent overfitting to any single task. Secondly, the model might benefit from observing shared features across tasks, which could contribute to a more robust understanding and better predictive performance. Lastly, and to a converse effect, multitask learning might saturating the model's capacity by having to share information across various tasks, leading to reduced performance on some tasks that require specialisation.

Q5: Using Results from Multiple Subject Systems

Assessors have the opportunity to learn from the evaluation results of multiple subject systems. But does that actually provide any benefit, e.g. in terms of regularisation? Training an assessor on the evaluation results of multiple subject systems may potentially yield regularisation benefits, but the empirical gains remain uncertain. To explore this, we trained an assessor using a diverse set of BIG-G model sizes in a population $M = \{4b, 8b, 27b, 128b\}$. To avoid test contamination, we ensured that all evaluation records associated with an instance were assigned to the same data split.

We used tags such as `"BIG-G 128b"` to prefix inputs, allowing the assessor to tailor its predictions according to the model family and size. Figure 5.9 contrasts the performance of this multi-subject assessor with a single-subject assessor trained exclusively on BIG-G 128b.In a second experiment, we further train the multi-subject assessor for 2 epochs on BIG-G 128b data exclusively, in order to specialise it again to that model, but from a pre-trained multi-subject starting point. Figure 5.10 shows those results.

In general, the addition of multiple subjects has a mixed effect on predictive accuracy, similar to the addition of multiple tasks (Section Q4), and again with sometimes sizeable differences. The extra fine-tuning in Figure 5.10 does not unlock a consistent effect either, although performance does slightly increase over the non fine-tuned version. Total AUROC of the multi-subject assessors does increase slightly over the single-subject reference suggesting limited benefit from pooling scores.

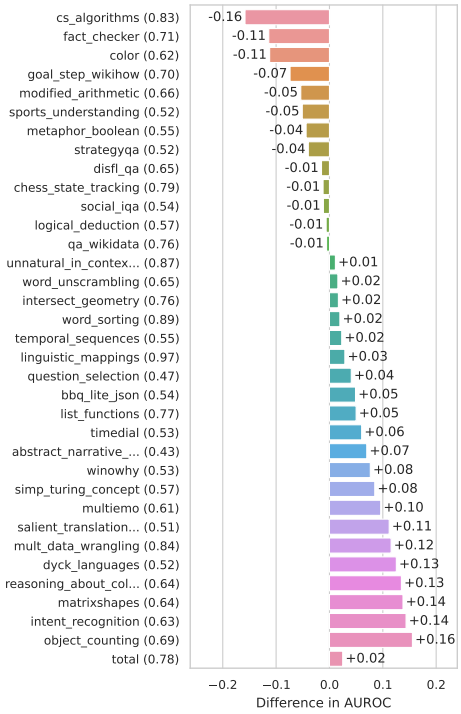


Figure 5.9: Single-subject vs. multi-subject assessors. Difference in AUROC between a single subject assessor trained on BIG-G 128b (from Q1) and a multi-subject assessor using evaluation results from the four best BIG-G models. Single subject assessor AUROCs are given in parentheses for each task, with bar charts showing multi-subject performance against this baseline. Performance is measured w.r.t. to predicting scores for BIG-G 128b only. (Q5)

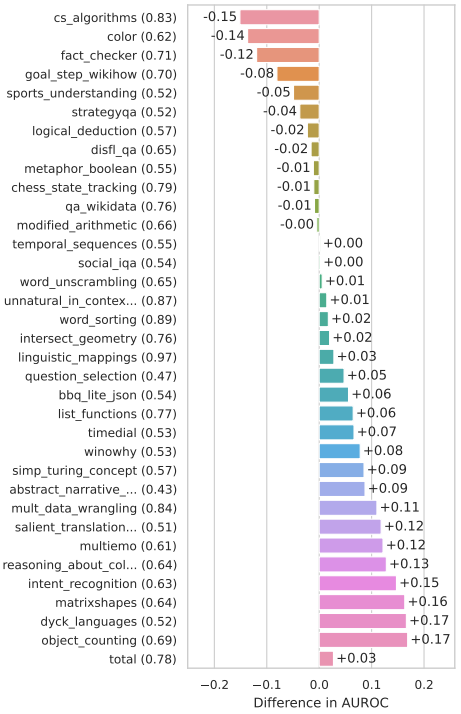


Figure 5.10: Single-subject vs, multi-subject assessors with fine-tuning. Comparison between a single subject (BIG-G 128b) assessor and a multi-subject assessor trained on four BIG-G models, focusing on BIG-G 128b scores. Unlike in Figure 5.9, the multi-subject is further trained for 2 epochs at the end of its training run on BIG-G 128b data only to specialise to this model.

A potential hypothesis is that for the tasks where performance improved, the subjects commonly agree on their answer, with the opposite holding for tasks where assessor performance decreases. In general the utility of population data is likely dependent on the similarity of the population to subject you wish to predict scores for.

Q6: Scaling in two directions

In this experiment, we investigate how the size of both the subject and assessor models affects predictability. We vary the size of the assessor between 44 million parameters (small), 86 million (base), which is the size we have been using for the other experiments, and 304 million (large). All pre-trained DeBERTa models are again provided by the HuggingFace Model Hub. We also use the evaluation results of all 12 BIG-G sizes available in BIG-bench, and for each subject size, we train an assessor of each of the three sizes, totalling $3 \times 12 = 36$ assessors.

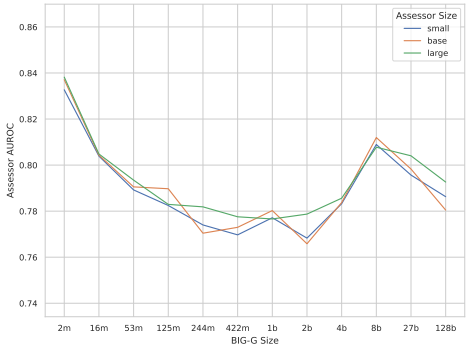


Figure 5.11: Performance (AUROC) of the assessor for various sizes of the subject and assessor. (Q6)

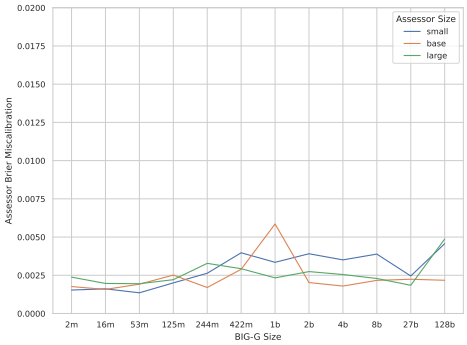


Figure 5.12: Brier score miscalibration (MCB) of the assessor for various sizes of the subject and assessor.

Figure 5.11 shows the performance (AUROC) of the assessor for various sizes of both the subject and the assessor. Each point represents a separate assessor. Interestingly, the size of the subject does not exhibit a clear relationship with predictability, which contrasts with results reported in (Kadavath et al., 2022, Fig. 33), where smaller models were found to be more predictable. One possible explanation for this discrepancy is that our assessors do not have access to the answers generated by the subject systems, whereas the models in (Kadavath et al., 2022, Fig. 33) do. This suggests that while the answers from smaller models might be poor in a detectable way, it is not necessarily easier to predict in advance for which instances they will fail.

Similarly, there is no clear advantage for using bigger assessors. Performance across all assessor size remains closely together, with no single size consistently taking the top. A possible explanation could be that the smallest assessor has sufficient capacity to model the problem, and better performance is currently bounded by a insufficient data rather than insufficient models.

Figure 5.12 complements the analysis by presenting the Brier score miscalibration (MCB) of the assessor for various sizes of both the subject and the a. Unlike AUROC, MCB does not show significant trends, likely because the calibration error remains close to zero across different subject and assessor sizes. Note the small scale of the y-axis, which only goes up to 0.02 miscalibration error, indicating generally low calibration errors.

Q7: Influence of Shots

Our final experiment analyses whether the presence of few-shot examples in the input impacts the assessor's performance. Modern language models benefit tremendously from having input-output examples or 'shots' present in their prompts. Both the subject system and the assessor receive the same input. When we refer to evaluation results containing k-shot instances, it means that the input includes k examples, which is the same for both the subject and the assessor. However, the few-shot examples that help the subject complete the task may not necessarily help the assessor, and act as distractors instead, or change the subject behaviour to be more unpredictable.

In this experiment we train four assessors with the BIG-G 128b evaluation results for 0, 1, 2 to 3-shot data respectively. As more examples are included, the percentage of prompts that exceed DeBERTa's 512-token input limit increases. This necessitates truncation of the input, either at the beginning (left) or the end (right). In the BIG-bench tasks, important information about the question might be either at the beginning or end of the input, with examples coming before, after, or in between task description. Consequently, truncating part of the input likely removes information crucial to the assessor, thereby impacting its performance. Moreover, it could also possible that few-shot prompting inherently results in less predictable subjects. The model behaviour might become more complex and harder to predict as the prompt length and structure become more intricate and longer.

However, the results in Figure 5.13 reveal that the number of shots has little influence, with total AUROCs varying between 0.77 and 0.79. The length of the prompt does not seem problematic, nor does the behaviour of the model become

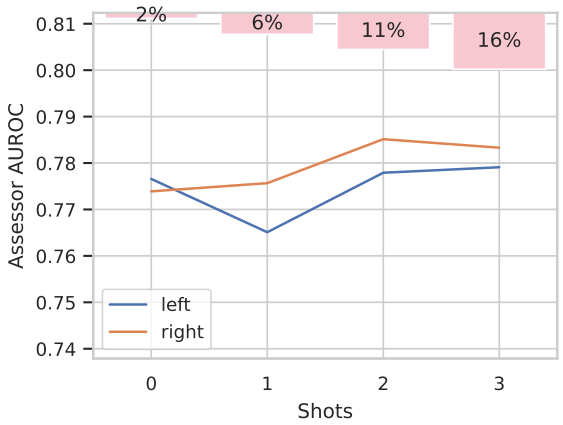


Figure 5.13: Trend of the assessor AUROC as more examples are added to the subject input, and thus also to the assessor input. The red bars represent the amount of questions that do not fit in the assessors 512 tokens maximum input length. Each point represents a separate assessor. Truncation side refers to from which side sequences are truncated if they are too long. Right (end) is the default in many language models, and is also what we use in the other experiments, but left (beginning) could makes for sense for assessors, depending on which part of the prompt containing the actual question instead of examples, as we expect that to be more informative.

more unpredictable with added examples. In fact, in the 2- and 3-shot settings, subjects seem slightly more predictable than in the 0- and 1-shot setting. The results also show that truncating from the left or right has negligible influence, with no big performance differences, and side being consistently better.

A possible explanation for the eclectic behaviour could be that the BIG-bench results we rely on have different instances and slightly different tasks across different shots. We also expect effects to be more pronounced in dis-aggregated from, i.e. split per task. We refrain from making any conclusions.

5.4 Discussion

Our experiments show that language model performance at the level of instances can be predictable (Q1), for some tasks to a significant extent, and that it can be predicted by other language models. This question, and its positive answer, is very relevant for a number of reasons:

1. Language models are general-purpose systems with complex **user interactions**. If we have tools that can anticipate when these models will succeed, we can significantly improve user interactions. Assessors can be integrated into user interfaces to refine prompts and other commands, leading to more efficient and effective interactions. See for example Martian[5], an LLM router that forwards each request to the best LLM.

2. The **modest training size** per task (e.g., the behaviour of a 128 billion parameter model can be learned with less than 1000 samples per task) highlights the high potential for incorporating language models into specific applications.These samples can be captured in just a few user interactions. The low number of samples required and the capability to nearly perfectly anticipate performance for some tasks are also relevant for research on knowledge localisation De Cao et al., 2021 and sparsity Fedus et al., 2022.

3. As the compute and training regimes scale up for language models, we can expect their predictability to remain similar. As shown in Fig. 5.11 and discussed in Q6, the gap between subject models and assessors does not widen with scale.

All this is reassuring for the predictability problem and the vision that very powerful systems of the future can be overseen by similarly sized systems which anticipate the validity of results.

A key finding is that *refined* out-of-distribution (OOD) performance prediction is difficult (Q2), but that there is potential for calibrated predictions, which is a significant finding that requires additional confirmation. With Q4 and Q5 we find that observing evaluations for multiple tasks and multiple subjects might provide some benefit for some tasks. If this can be improved in the future, finding ways to have more positive results for performance prediction, this would be very useful for the variety of use cases that derive from it like model routing or granular scaling laws Schellaert, Hamon, et al., 2024. The vision is that we can overturn the disaggregational analysis of language models and other tools. Instead of selecting and evaluating systems by aggregate performance and asking the question of why these measurements do not extrapolate to other distributions, we can think of a bottom-up approach, where we try to refine predictions at the instance level, aggregating and extrapolating from there to other examples, tasks and distributions.

[5]https://withmartian.com/

Concrete Limitations A couple of factors impact the utility and applicability of the results presented in this chapter. First, the tasks included in the BIG-bench evaluation suite are not entirely representative of those typically studied in the NLP community. Some example non-standard tasks include GEOMETRIC_SHAPES, CIFAR10_CLASSIFICATION or CHECKMATE_IN_ONE. While this is the nature of the BIG-bench effort, the applicability of the experimental findings could be improved by working with evaluation results from more established benchmarks, e.g. those provided by HELM Liang et al., 2022. Additionally, in terms of statistical significance, one must note there are no repetitions of the experiments and no validation set was used for early stopping and hyperparameter selection.

Another limitation is that text-to-text models have not been studied as assessors, only as subject systems. While language models have seen significant increases in scale, the assessor models we train are limited in this regard, with a maximum size of 320 million parameters.

These experiments were relatively sensitive to the perturbations in the data or the exact assessor setup, and would benefit significantly from additional repetitions to draw more confident conclusions. The surprisingly good OOD calibration performance especially bears repeating, and further investigation in general.

Lastly, and as mentioned before: the BIG-G models that act as subjects have not been fine-tuned or trained directly on any of the tasks they are evaluated on, while the assessors have been. Their self-assessment, based on the confidence in their output, could thus equal or beat the assessor in score prediction after fine-tuning on the used tasks. This difference should be taken into account when making or interpreting claims about potential assessor benefits. Additionally, BIG-G models have quickly become out-dated, giving the fast pace of the field and the countless new models that are released each month. It is not clear how representative these models still are.

Chapter 6

Human Score Predictors

Millions of people are using general-purpose AI systems based on LLMs, becoming commonplace in areas such as education (Kasneci et al., 2023), medicine (Thirunavukarasu et al., 2023), science (Birhane et al., 2023; Messeri & Crockett, 2024) and administration (Eloundou et al., 2023; G. Kim et al., 2023). As these models frequently make mistakes, users have to supervise model operation and manage their expectations of performance, for the sake of a reliable and efficient usage. Is there a decent enough chance to get a correct answer? Should the answer be validated? Should the query be reformulated to get better results? For example, human expectations of subject success depend on their perception of the difficulty of instances (Bansal et al., 2019, 2021; Feather, 2021), but is that expectation warranted?

With language models becoming larger and more instructable, we are also urged to analyse how this predictability has evolved over time. Since the very early LLMs (Brown et al., 2020; Devlin et al., 2019; Raffel et al., 2020), models have been *scaled up* —trained with more parameters, on larger datasets, and with longer training times—, and have also been *shaped up* with human feedback —using techniques such as instruction fine-tuning (Chung et al., 2022), Reinforcement Learning from Human Feedback (RLHF) (Ouyang et al., 2022), or output filtering moderation techniques (Markov et al., 2023; OpenAI, 2023c).

It may be taken for granted that as models become more powerful and better aligned by using these strategies, they also become more reliable from a human perspective, i.e., that their errors follow a predictable pattern that humans can understand and adjust their queries to (Schellaert et al., 2023). For instance, early models failed at simple additions such as "20 + 183". Performance was highly predictable: failure was common. As a result, users easily understood there was no operating range for this task: nobody used these models for addition. A few scaled-up shaped-up generations later, the models not only seemingly master these additions but even succeed with additions of 50 digits or more. Because of this prowess, people may start using them as calculators (e.g., converting measurements to different units (Tsigaris & Teixeira da Silva, 2023)). It is only now that users become bitterly disappointed when the model fails at a simple prompt such as "`Add 3913 and 92`". The *user-driven* reliability is then seriously damaged, since the model fails when the user thought they were in operating range. The experience becomes even more baffling when the user gets the correct answer with a slight twist in the question such as "`3913 + 92 =`", or with no change at all—since many models are configured to be non-deterministic. While this prompt sensitivity has been analysed extensively (P. Liu et al., 2023; Sanh et al., 2021; J. Sun et al., 2023; Y. Wang et al., 2022), it is poorly understood why an over-diligent system spouts a wrong answer for a 100-digit addition instead of simply answering "*I'm afraid I can't do that*". This reckless behaviour has been incentivised by developers building models that are "never evasive" (Bai et al., 2022).

6.1 Research Questions

We investigate three core and intertwined elements that affect the predictability of LLMs from a human perspective. Each question will be analysed considering both an evolution of the corresponding metrics over increasing difficulty, and over the evolution of models. The former because difficulty is a feature users rely on in the management of their expectations, the latter so we can analyse how predictability has evolved as the field has advanced.

1. **Difficulty concordance**: Are errors more likely for items that humans perceive as difficult? Do scaling and shaping eliminate errors for easy items, therefore creating areas of reliable operation? *Difficulty concordance is important because, if human difficulty would be highly indicative of subject performance, users would an high-level instance feature available that is interpretable, general, and easily accessible.*

2. **Task avoidance**: How often do language models give plausible but wrong answers instead of safely avoiding answering questions? Are scaled-up shaped-up models better

at avoiding errors or making them detectable by humans? *Avoidance is of interest because it reduces the need for predictability from a safety perspective. If subjects have adequate self assessment, and tell the user when they do not know the answer, there is less risk of over-trust, and no more need for validation of the answer.*

3. **Prompting stability**: How are correctness and avoidance affected by tangential changes in the prompt? Are scaled-up shaped-up models less sensitive to prompt variation across difficulties? *Prompt sensitivity acts as an upper bound to predictability. Having seen a correct answer on an identical question, just slightly reworded, should provide significant evidence that the question will be answered correctly in the future. High prompt sensitivity spells bad news for accurate models of performance, as there is no way to collect more relevant information.*

We will answer these questions by using human difficulty metrics for carefully selected benchmarks (see table 6.2), examining different kinds of avoidance (table C.10), and using 15 natural prompt variations —questions conceived by humans as meaning the same— per benchmark (table C.1).

Difficulty, avoidance, prompting, and their evolution have been analysed from different angles (L. Chen et al., 2024; Kadavath et al., 2022; L. Kuhn et al., 2022; Sanh et al., 2021; J. Sun et al., 2023; Y. Wang et al., 2022; H. Zhang et al., 2024). Appendix C.11 provides a full discussion. Here, we uniquely focus on the systemic interaction of these three elements, and their evolution as LLMs have been scaled and shaped up, and their relation to predictability.

Because the experimental setup is extensive, it is presented in three progressively more detailed sections. First, here below, then in section 6.5 on methods at the end of this chapter, and finally in the full Supplementary Information chapter (appendix C), which also contains additional analysis.

6.2 Experiment Setup

We analyse the trajectory of several families of LLMs: OpenAI's GPT saga, Meta's LLaMA series and BigScience's BLOOM suite. GPT has led the state of the art in the past few years, and according to several surveys (Bommasani et al., 2023; J. Yang et al., 2024; W. X. Zhao et al., 2023), is central to the LLM ecosystem, influencing transformer-based architectures, training data, evaluation frameworks and alignment techniques. LLaMA (Touvron, Martin, et al., 2023) is the best example of a family whose weights have been released, and has therefore taken up a prominent role in the ecosystem due to its countless fine-tunes and

adaptations, while BLOOM (BigScience et al., 2023; Muennighoff et al., 2023) is the result of an even more open endeavour coming from the scientific community.

Each family represents a genuine effort of making LLMs more capable and better aligned at the same time. Table 6.1 summarises the details of the models in these three families. Scaling (increasing parameters, data and thus compute) has been identified as a key predictor for overall performance (Kaplan et al., 2020), and shaping (modifying the trained systems) has improved their instructability and alignment. This creates two categories: the 'raw' models—GPT-3 Ada, Babbage, Curie and Davinci, the non-chat LLaMA models and the base (non-z) BLOOM models—contrasted against the 'shaped-up' models, also referred to as 'instruct' or 'chat' models, which include some kind of 'instruction adaptation' (W. X. Zhao et al., 2023), fine-tuning, or safety moderation of the outputs. For our analysis, it will prove very convenient that BLOOM and LLaMA have 6 and 3 exactly *paired* versions, respectively, of raw and shaped-up models, when disentangling scaling up from shaping up.

We identify good intrinsic proxies for human difficulty based on relevant literature in the first two domains (addition and anagram), or by identifying demand-related features in the rest (excluding science, for which multiple human difficulty assessments were already available for all the instances (Mihaylov et al., 2018)). To determine their quality, we conducted an extensive human study S1 to assess which difficulty proxies best matched human expectations, and calibrate the proxies to a normalised difficulty score, ranging from 0 to 100, representing the anticipated percentage of failure for the 'average human'. Systematically controlling for human difficulty is not only novel in the analysis of LLMs, but is crucial for the understanding of user-driven predictability: human expectations of success depend on their perception of the difficulty of instances (Bansal et al., 2019, 2021; Feather, 2021). Table 6.2 provides an overview of the five benchmarks, the intrinsic difficulty function used as a proxy for human difficulty (discussed in the Methods section), some examples, and the calibrated human difficulty values for the given examples.

Another necessary and innovative element in our analysis is that we consider three categories for the responses: correct, incorrect and 'avoidant', denoted by **c**, **i** and **a**, respectively. Avoidance in human subjects has been extensively explored in psychology (Covington, 1984; Marecek & Mettee, 1972; Zeidner et al., 2005). Such avoidant behaviours include procrastination, deviation, making excuses, or simply not answering. For LLMs, avoidance is also referred to as 'hedging', 'refusal' (OpenAI, 2023c) or 'evasiveness' (Bai et al., 2022), including fortuitous utterances or continuations that are not answers (non-conforming), and those responses at the meta-level explaining why the question is not answered (for epistemic or

ethical reasons). Table C.10 shows types of avoidance for some tasks in the five benchmarks.

Difficulty concordance, task avoidance and prompting stability must be regarded from the point of view of human users interacting with LLMs. Our human study S1 (see appendix C.6) analyses whether human perceptions of difficulty in general are aligned with actual human performance and self-confidence, because this has important implications in the tasks humans decide to delegate to language models and their prompt formulation. But as crucial as the inputs are, so is the way the *outputs* from the model are used, verified or supervised. The context of use of both inputs and outputs determines how reliable the use of these systems is. We conducted a second human study S2 (see appendix C.7), where we explore whether humans can assess models' outputs accurately and thus compensate for different types of error. With a three-valued confusion matrix with correctness, avoidance and incorrectness, we can focus on the frequency of non-avoidant cases where humans believe the output is correct but it is not (fig. 6.5).

Data & Code

All data, including existing and newly created datasets, prompts, model responses, grading (manual and automatic), the human study data (questions and responses) are available at a Zenodo archive (L. Zhou et al., 2024a), as well together with the code in a Git repository https://github.com/wschella/llm-reliability. To hinder data contamination from automated web scraping, the relevant data files are behind a password-encrypted zip file, whose access code is also provided in the repository.

All code, including for data analysis, human study, plotting, algorithmic grading conditions, and interacting with language models is available at the same Zenodo archive and Git repository.

Table 6.1: Overview of GPT, LLaMA and BLOOM models. The ten GPT, ten LLaMA and twelve BLOOM models we analyse in this paper, including their release year, scaling factors (#parameters, data size, compute) and shaping techniques (fine-tuning, use of human feedback, moderation, etc.). Key abbreviations include: FeedME (a supervised fine-tuning method using human-written demonstrations and top-quality model samples), PPO (a reinforcement learning approach with reward models trained through human comparisons), RBRMs (rule-based reward models that enhance GPT-4 policy model by promoting safe content generation and discouraging harmful outputs during RLHF fine-tuning), FT (Fine-Tuning), CD (Context Distillation) and RS (Rejection Sampling), preceded with 'S-' if aligned for safety. The sources of the specs in the table can be found in appendix C.1.

Model	Year	Scaling			Shaping	
		Size (#params)	Data (#tokens)	Compute (#FLOPs)	Instruction	Alignment
GPT-3 Ada	2020	350M	300B	6.41e+20	None	None
GPT-3 Babbage	2020	1.3B	300B	2.38e+21	None	None
GPT-3 Curie	2020	6.7B	300B	1.20e+22	None	None
GPT-3 Davinci	2020	175B	300B	3.14e+23	None	None
text-davinci-001	2021	175B	–	–	FeedME	None
text-davinci-002	2022	175B	–	–	FeedME	None
text-davinci-003	2022	175B	–	–	RLHF (PPO)	None
GPT-3.5-turbo	2022	175B[1]	–	–	RLHF[2]	S-FT, Moderation
GPT-4 v1	2023	–	–	–	RLHF[2]	S-RLHF, RBRMs, Moderation
GPT-4 v2	2023	–	–	–	RLHF[2]	S-RLHF, RBRMs, Moderation
LLaMA-7b	2023	6.7B	1.0T	4.02e+22	None	None
LLaMA-13b	2023	13B	1.0T	4.55e+22	None	None
LLaMA-33b	2023	32.5B	1.4T	2.73e+23	None	None
LLaMA-65b	2023	65.2B	1.4T	5.50e+23	None	None
LLaMA-2-7b	2023	7B	2.0T	8.40e+22	None	None
LLaMA-2-13b	2023	13B	2.0T	1.60e+23	None	None
LLaMA-2-70b	2023	70B	2.0T	8.10e+23	None	None
LLaMA-2-7b-chat	2023	7B	2.0T	8.40e+22	RLHF (PPO & RS FT)	S-FT, S-RLHF & S-CD
LLaMA-2-13b-chat	2023	13B	2.0T	1.60e+23	RLHF (PPO & RS FT)	S-FT, S-RLHF & S-CD
LLaMA-2-70b-chat	2023	70B	2.0T	8.10e+23	RLHF (PPO & RS FT)	S-FT, S-RLHF & S-CD
BLOOM-560m	2022	559M	350B	1.83e+21	None	None
BLOOM-1b1	2022	1.07B	350B	3.60e+21	None	None
BLOOM-1b7	2022	1.72B	350B	5.57e+21	None	None
BLOOM-3b	2022	3.00B	350B	9.83e+21	None	None
BLOOM-7b	2022	7.07B	350B	2.32e+22	None	None
BLOOM-176b	2022	176.25B	366B	5.77e+23	None	None
BLOOMz-560m	2022	559M	353.67B	1.87e+21	Multitask FT	None
BLOOMz-1b1	2022	1.07B	350.5B	3.69e+21	Multitask FT	None
BLOOMz-1b7	2022	1.72B	358.4B	5.70e+21	Multitask FT	None
BLOOMz-3b	2022	3.00B	358.4B	1.00e+22	Multitask FT	None
BLOOMz-7b	2022	7.07B	354.2B	2.38e+22	Multitask FT	None
BLOOMz-176b	2022	176.25B	368B	5.91e+23	Multitask FT	None

[1] It is understood that GPT-3.5-turbo is an improvement on text-davinci-003 as described in OpenAI (2023b), and thus should have 175B parameters, but this information is not explicitly declared by OpenAI in any official sources. [2] Specific method unknown.

Table 6.2: The five benchmarks. With examples, chosen difficulty metric, and calibrated difficulty values according to human expectations (% failure).

Benchmark	Examples	Cal. Diff.
addition — single-task benchmark Arithmetic operations ranging from one to one-hundred-digit additions. *Difficulty: #carrying operations (f_{cry})*	`Make the addition of 24427 and 7120.`	35.25
	`The sum of 47309068053 and 95464 is`	65.04
	`189360301032350163843O + 98832380858765261900 =`	98.67
anagram — single-task benchmark Jumbled words to be unscrambled to form a meaningful word ranging from three to twenty-letter words. *Difficulty: #letters of anagram (f_{let})*	`Unscramble this string of letters, "efe", to form a word.`	18.42
	`Rearrange the letters "ngiotuq" to make a single word.`	50.42
	`Rearrange the following anagram into an English word: "elmtweoascnednkg".`	96.78
locality — single-task benchmark Geographical knowledge about the location and size of cities relative to each other. *Difficulty: Inverse city popularity (f_{pop})*	`Which city that is less than 27 km away from Toronto has the largest number of people?`	91.66
	`What is the name of the largest city (by population) that is less than 98 km away from Altea?`	92.64
	`Name the most populated city that is less than 39 km away from Akil.`	99.87
science — multi-task benchmark Elementary science-related world knowledge questions and graduate-level questions in biology, physics, and chemistry. *Difficulty: Anticipated human difficulty (f_{hum})*	`Definition: In this task, you need to provide the correct option for a given problem from the provided options.\nProblem: Shining a light through a diamond can \nA) make a lot of bright lights shine\nB) summon a brilliant wave of color\nC) heat up a room\nD) make a lot of money\nOutput:`	37.02
	`A light beam is propagating through a glass with index of refraction n. The glass is moving at constant velocity v in the same direction as the beam and toward the observer in laboratory. What is the speed of light in glass relative to the observer in laboratory? Take the speed of light in vacuum c=1.\nA. (1+n*v)/(n+v)\n B. (1-n*v)/(n+v)\n C. 1 D. (1+n*v)/(n-v)\nWith respect to the choices above, the correct one is`	71.83
	`Answer the following questions based on the list of available choices\nIdentify the missing reagents in the following reaction.\n(3r,5r,7r)-adamantane-1-carboxylic acid + A --> (3r,5r,7r)-adamantane-1-carbonyl azide + B --> (3s,5s,7s)-adamantan-1-amine.\nA: A = NaN3 and B = HCl aq, Heat\nB: A = PCl5 and B = H3O+, Heat\nC: A = diphenylphosphoryl azide (DPPA) and B = H3O+, Heat\nD: A = diphenylphosphoryl azide (DPPA) and B = NaN3\nAnswer:`	99.97
transforms — multi-task benchmark Information-centric transformation tasks. *Difficulty: Combination of input+output word count and Levenshtein distance (f_{w+l})*	`Be concise in your answer, placed between double quotes. Do not generate any explanation or anything else apart from the requested output. Given\n"double07@MI6.gov.uk"\nModify the input to display the domain of the email address of the form USER@DOMAIN.`	39.49
	`Consider the INPUT: \n"8:30h - Accreditation (badges)\n9:00h - Opening\n9:15h - Keynote\n10:15h - Coffee break\n10:45h - Invited Talks\n11:55h - Lightning talks\n12:05h - Panel\n13:00h - Lunch break (in the hall)\n14:30h - Keynote\n15:30h - Minibreak\n15:40h - Invited Talks\n16:50h - Panel\n17:45h - Closing remarks"\nI'd like the agenda to show a 15-minute reduction in each keynote speaker's segment, shifting the schedule to finish earlier. \nBe concise in your answer, placed between double quotes. Do not generate any explanation or anything else apart from the requested output.`	55.22
	`Michael Vaughn, a 63-year-old retired naval officer, presents an extensively complex medical history complicated by a litany of allergies. He battles chronic pain stemming from neuropathy for which he takes Pregabalin (Lyrica) 150 mg twice daily. Due to advanced rheumatoid arthritis, he relies on Etanercept (Enbrel) 50 mg, administered weekly via subcutaneous injection, but cannot be prescribed common NSAIDs like Ibuprofen or Naproxen due to gastrointestinal bleeding and a reported severe allergy to Aspirin (anaphylaxis). His Type 2 diabetes is managed with Insulin Aspart (NovoLog) administered via an insulin pump with doses varying according to his blood glucose readings; he experienced a life-threatening lactic acidosis episode with Metformin.\n I'd like the list of drugs that are prescribed to the patient to be arranged alphabetically and without repetitions, in the form of a clean, comma-separated list. Be concise in your answer, placed between double quotes. Do not generate any explanation or anything else apart from the requested output.`	64.76

6.3 Results

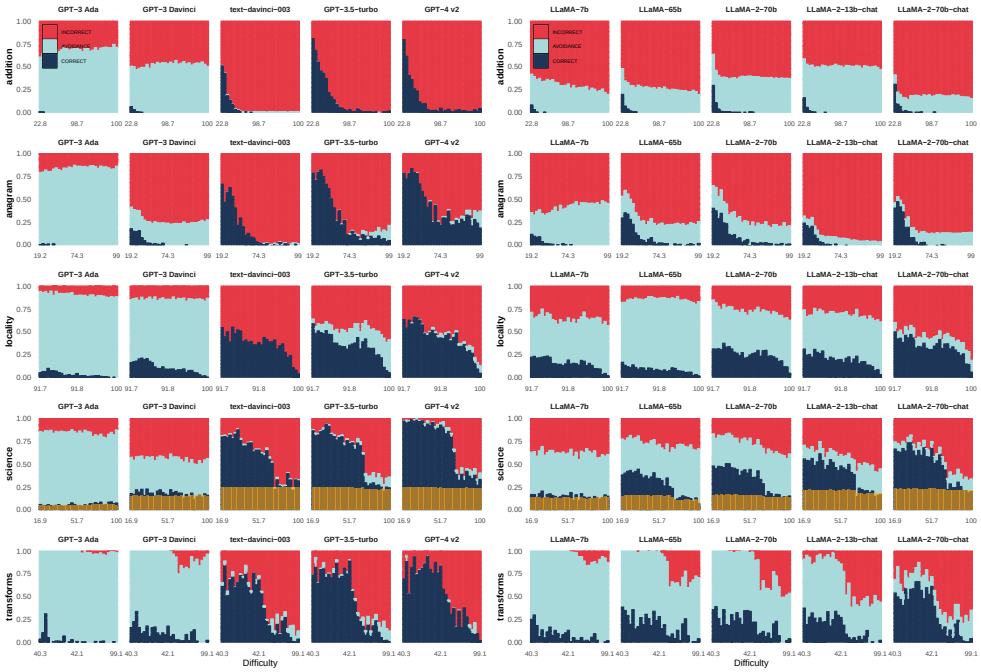


Figure 6.1: Performance of a selection of GPT and LLaMA models over difficulty. The values are split by correct, avoidant and incorrect results. For each combination of model and benchmark, the result is the average of 15 prompt templates, listed in table C.1. For each benchmark, we show its chosen intrinsic difficulty, monotonically calibrated to human expectations on the x-axis for ease of comparison between benchmarks. The x-axis is split into 30 equal-sized bins, whose ranges must be taken as indicative of different distributions of perceived human difficulty across benchmarks. For science the transparent yellow bars at the bottom represent the random guess probability (25% of the non-avoidance answers). Plots for all GPT and LLaMA models can be found in Supplementary Information figs. C.13 and C.14, where the same plot for BLOOM (fig. C.15) can also be found.

6.3.1 Performance

Over Models

Figure 6.1 shows the results of a selection of models in the GPT and LLaMA families, increasingly scaled-up and shaped-up, for the five domains: addition, anagram, locality, science and transforms. Note that both scaling and shaping are displayed

on a single axis. We see that the percentage of correct responses increases for scaled-up shaped-up models, as we approach the last column. This is an expected result and holds consistently for the rest of the models, shown in fig. C.13 (GPT), fig. C.14 (LLaMA), and fig. C.15 (BLOOM).

Over Difficulty

Let us focus on the evolution of correctness with respect to difficulty. For addition, we use the number of carry operations in the sum (f_{cry}); for anagram, we use the number of letters of the given anagram (f_{let}); for locality, we use the inverse of city popularity (f_{pop}); for science, we use human difficulty directly (f_{hum}); and for transforms, we use a combination of input and output word count and Levenshtein distance ($f_{\text{w+l}}$) (see table 6.2). As we discuss in the Methods section, these are chosen as good proxies of human expectations about what is hard or easy according to the human study S1 (see appendix C.6). As difficulty increases, correctness noticeably decreases for all models. Table C.7 measures the correlations between correctness and the difficulty function. With the except of BLOOM for addition, all of them are high (> 0.78).

Yet despite the predictive power of human difficulty metrics for correctness, full reliability is not even achieved at very low difficulty levels. While the models can solve very challenging instances, they also still fail at very simple ones. This is very evident for anagram (GPT), science (LLaMA), locality and transforms (GPT and LLaMA), proving the presence of a difficulty discordance phenomenon. The discordance is observed across all LLMs, with no apparent improvement through the strategies of scaling up and shaping up, confirmed by the aggregated metric in fig. 6.4. This is especially the case for GPT-4, in comparison to its predecessor GPT-3.5-turbo, primarily increasing performance on instances of medium or high difficulty with no clear improvement on easy ones. For the LLaMA family, no model achieves 60% correctness at the simplest difficulty level (discounting 25% random guess for science). The only exception is a region with low difficulty for science with GPT-4, with almost perfect results up to medium difficulties.

6.3.2 Avoidance

Over Models

Focusing again on the trend across models, we also see something more: the percentage of incorrect results increases dramatically from the raw to the shaped-up models, as a consequence of substantially reducing avoidance (almost disappear-

ing for GPT-4). Where the raw models tend to give odd non-conforming outputs that cannot be interpreted as an answer (fig. C.12), shaped-up models instead give seemingly plausible but wrong answers. More concretely, the area of avoidance in fig. 6.1 decreases drastically from GPT-3 Ada to text-davinci-003 and is largely replaced with more incorrect answers. Then, for GPT-3.5-turbo, avoidance increases slightly, only to taper off again with GPT-4. This trade from avoidant to incorrect answers is less pronounced for the LLaMA family, but still clear when comparing the first versus the last models. This is summarised by the prudence indicators in fig. 6.4, showing the shaped-models perform worse in terms of avoidance, which does not match the expectation that more recent LLMs would more successfully avoid answering outside their operating range. In our analysis of the types of avoidance (see appendix C.13), we do see non-conforming avoidance changing to epistemic avoidance for shaped-up models, which is a positive trend. But the pattern is not consistent, and cannot compensate for the general drop in avoidance.

Over Difficulty

Looking at the trend over difficulty, the important question is whether avoidance increases for more difficult instances, as would be appropriate for the corresponding lower level of correctness. Figure 6.1 also shows this is not the case. There are only a few pockets of correlation and the correlations are weak. This is the case for the last three GPT models for anagram, locality and science and a few LLaMA models for anagram and science. In some other cases, we see an initial increase of avoidance but then stagnation at higher difficulties. The percentage of avoidant answers rarely rises quicker than the percentage of incorrect ones. The reading is clear: errors still become more frequent. This represents an involution in supervisability: **there is no difficulty range where errors are unlikely because the questions are so easy the model never fails, and neither is there a range where they are so difficult the model always avoids an answer**.

6.3.3 Sensitivity

Over Models

Would it be possible that this lack of reliability may be motivated by some prompts being especially poor or brittle, while still finding a secure region for some particular prompts? We now analyse prompt sensitivity by disaggregating correctness, avoidance and incorrectness, across the different used prompts (listed

in table C.1). As depicted in fig. 6.2 for the most representative models of the GPT family, we observe a big difference between the 'raw' models, represented by GPT-3 Ada and Davinci, and the other models of the GPT family. Shown in fig. 6.3, the LLaMA family underwent a more timid, but nonetheless noticeable transformation, as exemplified by LLaMA-2 70-b chat and it's reduced sensitivity. The 'raw' GPT and all the LLaMA models are very sensitive to the prompts, even in the case of highly unambiguous tasks such as addition. An alternative disaggregation can be found in fig. C.1, confirming that shaped-up models are, in general, less sensitive to prompt variation.

Over Difficulty

Difficulty does not seem to affect sensitivity very much, and for easy instances we see that the raw models (GPT-3 Davinci and non-chat LLaMA models in particular) have some capacity that is *only* unlocked by carefully-chosen prompts. Things change substantially for the shaped-up models, the last six GPT models and the last three LLaMA (chat) models, which are more stable, but with pockets of variability across difficulties.

Overall, these different levels of prompt sensitivity across difficulty have very important implications for users, as it implies performance can not be reasonably predicted, as even slight rephrasing of a question can flip the answer from correct to incorrect.

6.3.4 Summary Statistics

Figure 6.4 represents how some **key indicators show that the 'shaped-up' models (in blue) are more stable to prompt variation and are more correct, at the cost of being less concordant with human difficulty, and having more failures overall (less prudent)**. The indicators summarise the behaviour of five carefully-selected benchmarks in the domains of simple numeracy (addition), vocabulary reshuffle (anagram), geographical knowledge (locality), diverse scientific skills (science) and information-centric transformations (transforms). This covers a range of domains and degrees of openendedness of the answers.

6.3.5 Error Correction

Lack of predictability is not compensated by either model avoidance (as seen section 6.3.2), but also not by human supervision spotting the errors. This can be seen in fig. 6.5. Looking at the correct-to-incorrect type of error (red), if the user expectations on difficulty were aligned with model results, we should have fewer cases on the left area of the curve (easy instances), and those should be better verified by humans. This would lead to a safe haven or operating area on those instances that are regarded as easy by humans, with low error from the model and low supervision error from the human using the response from the model. But, unfortunately, this only happens for addition for very easy additions, and anagram for a wider range, since verification is easy in general.

6.3.6 Shaping versus Scaling

The observations so far on GPT and LLaMA also apply to the BLOOM family (fig. C.18), and to disentangle the effects of scaling and shaping, we conduct an ablation study using LLaMA and BLOOM models in their shaped-up versions (named 'chat' and 'z' respectively) against the raw versions, with the advantage that each pair has equal pre-training data and configuration. We also include all other models with known compute, such as the non-instruct GPT models.

We take the same data that is summarised in fig. 6.4 (C.11) and perform a scaling analysis using the FLOPs column from table 6.1, which usually captures both data and parameter count if models are well dimensioned (Hoffmann et al., 2022). We separate the trends between raw and shaped-up models. The fact that correctness increases with scale has been systematically shown in the literature of scaling laws (Hoffmann et al., 2022; Kaplan et al., 2020). But, with our data and three-outcome labelling, we can analyse the unexplored evolution of avoidance and incorrectness. This is shown in fig. 6.6 (left).

Clearly, as we have observed already, avoidance is much lower for shaped-up models (blue) than raw models (orange) but incorrectness is much higher. But even if correctness increases with scale, incorrectness does not decrease, and for the raw models, it increases considerably. This is a big surprise, and becomes more evident when we analyse the percentage of incorrect responses for those that are not correct in ($\mathbf{i}/(\mathbf{a}+\mathbf{i})$ in our notation) in fig. 6.6 (right). Here we see an alarming increase of the proportion of errors, with models becoming more 'ultracrepidarian' (increasingly giving a non-avoidant answer when they do not know, so failing proportionally more).

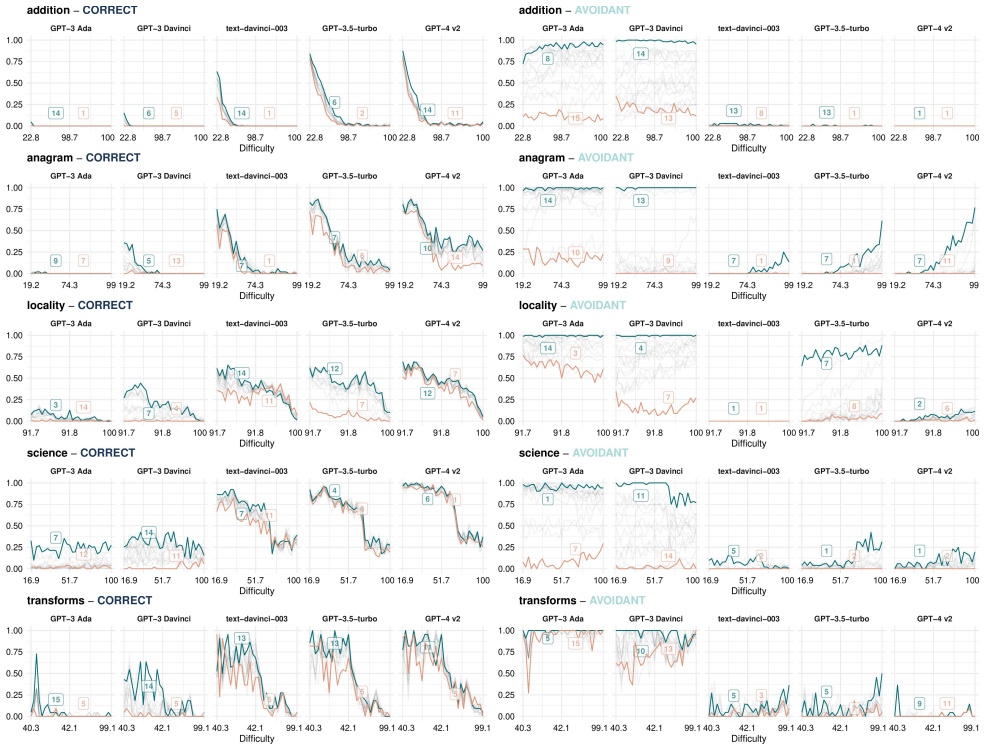


Figure 6.2: Prompting stability of GPT models over difficulty. Proportion of *correctness* and *avoidance* represented as (grey) curves over difficulty for the 15 prompt templates for the GPT models addressing each of the five benchmarks. The **green** and **bronze** curves correspond to the prompt template that has, respectively, the highest and lowest average correctness, avoidance, or incorrectness. The two small numbers in green and bronze in the plot identify them (corresponding to the template codes in table C.1). The plots for all the models, including BLOOM, and all response categories are in appendix C.15.

Figure 6.3: Prompting stability of LLaMA models over difficulty. Proportion of *correctness* and *avoidance* represented as (grey) curves over difficulty for the 15 prompt templates for the LLaMA models addressing each of the five benchmarks. Details as in fig. 6.2. The plots for all the models, including BLOOM, and all response categories are in appendix C.15.

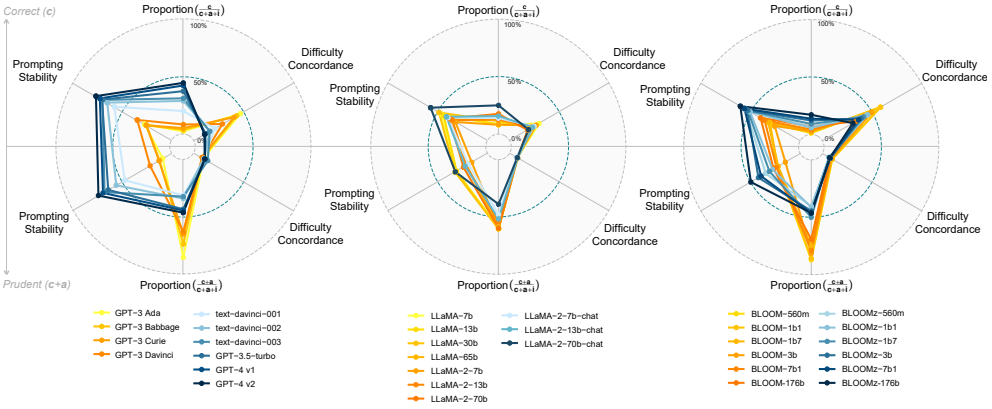


Figure 6.4: Supervisability indicators for the OpenAI's GPT, Meta's LLaMA and BigScience's BLOOM families. The 'raw' models (yellow to orange) and the 'shaped-up' models (light to dark blue) cluster differently. Since the answers for all these models fall into three categories (correct, avoidant and incorrect), shortened as **c**, **a** and **i** respectively, we have indicators for 'correctness' versus avoidance+incorrectness, and 'prudence' (correctness+avoidance) versus incorrectness. Looking at the correctness indicators (top half), which represent accurate responses, we see the 'shaped-up' models are more stable to prompt variations, are more frequently correct (higher correctness proportion), but less concordant with human difficulty than the 'raw' counterparts. Looking at the prudence indicators (bottom half), we see the new 'shaped-up' models are also more stable to prompt variations, but failing more frequently (lower prudence proportion, by avoiding less), and not much more concordant with human difficulty. Focusing on the 'shaped-up' models only (in blue), we observe that the most powerful GPT-4 v2, LLaMA-2-70b-chat and BLOOMz-176b perform best in correctness proportion and prompting stability (top and bottom), but equal or worse than other models for all the other indicators, with many fluctuations that do not indicate a clear positive trend in these other dimensions. Details of the indicators and data used for this plot are found in the Methods section. The full C.11 provides a more detailed perspective on the same results.

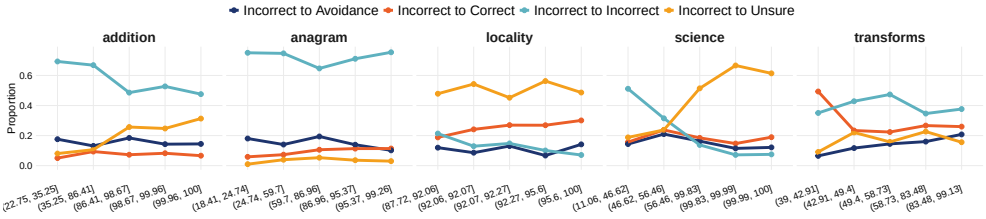


Figure 6.5: Evolution of types of supervision errors versus difficulty according to human survey S2. In the survey (fig. C.4), humans have to determine if the output of a model is correct, avoidant or incorrect (or do not know, represented by the 'Unsure' option in the questionnaire). Difficulty in the x-axis shown in equal-size bins. We see very few areas where the dangerous error (incorrect being considered correct by the human) is sufficiently low to consider a safe operating region.

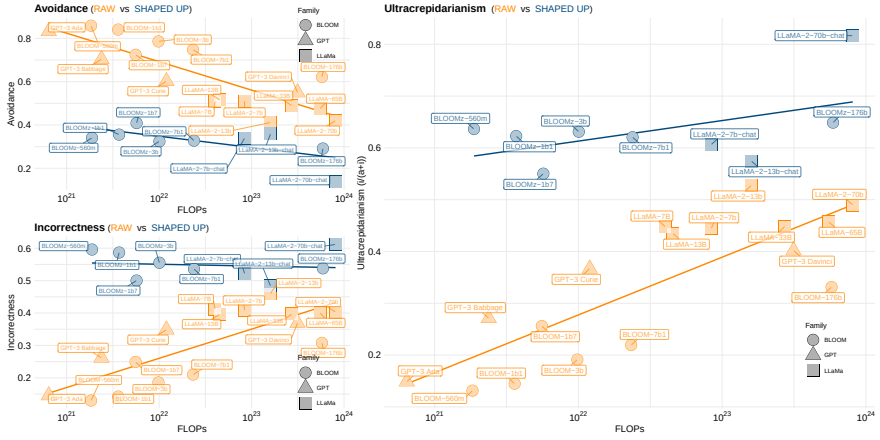


Figure 6.6: Scaling analysis of LLaMA and BLOOM families and the non-instruct GPT models. The plot employs a logarithmic scale for FLOPs. The focus is on avoidance (**a**, top left), incorrectness (**i**, bottom left) and ultracrepidarianism (**i**/(**a** + **i**), right) —the proportion of incorrect over both avoidant and incorrect answers.

6.4 Discussion

We can now take all these observations and trends into account, the expectations of a regular human user, as explored by human study S1, and the limited human capability for verification and supervision seen in study S2. This leads to a re-understanding of the supervisability evolution of LLMs, organised in groups of two findings for difficulty discordance ($F1_a$, $F1_b$), task avoidance ($F2_a$, $F2_b$) and prompt sensitivity ($F3_a$, $F3_b$):

$F1_a$ **Human difficulty proxies serve as valuable predictors for LLM correctness**: proxies of human difficulty are negatively correlated with correctness, implying that, for a given task, humans themselves can have approximate expectations for the correctness of an instance.

> **Relevance**: *This predictability is crucial as alternative success estimators when the model's self-confidence is either not available or significantly weakened (e.g., RLHF ruining calibration (M. Li et al., 2024; OpenAI, 2023c)).*

$F1_b$ **Improvement happens at hard instances while problems with easy instances persist, extending the difficulty discordance**: current LLMs clearly lack easy operating areas with no error. In fact, the latest models of all families are not securing any reliable operating area.

> **Relevance**: *This is especially concerning in applications that demand the identification of operating conditions with high reliability.*

$F2_a$ **Scaling and shaping currently trade avoidance for more incorrectness:** the level of avoidance depends on the model version employed, and in some cases, it vanishes entirely, with incorrectness taking important proportions of the waning avoidance (i.e., 'ultracrepidarianism').

> **Relevance**: *This elimination of the buffer of avoidance (intentionally or not) may lead users to experience initial overtrust in tasks they do not command, but being let down in the long term.*

$F2_b$ **Avoidance does not increase with difficulty, and rejections by human supervision do not either:** model errors increase with difficulty but avoidance does not. Users can recognise these high-difficulty instances but still make frequent incorrect-to-correct supervision errors.

> **Relevance**: *Users do not sufficiently use their expectations on difficulty to compensate for increasing error rates in high-difficulty regions, indicating overreliance.*

$F3_a$ **Scaling up and shaping up may not free users from prompt engineering**: our observations indicate that there is an increase in prompting stability. However, models differ in their levels of prompt sensitivity, and this varies across difficulty.

> **Relevance**: *Users may struggle to find prompts that benefit avoidance over incorrect answers. Human supervision does not fix these errors.*

$F3_b$ **Improvement in prompt performance is not monotonic across difficulty levels**: some prompts do not follow the monotonic trend of the average, being less conforming with the difficulty metric, and having fewer errors for hard instances.

> **Relevance**: *This non-monotonicity is problematic because users may be swayed by prompts that work well for difficult instances but at the same time get more incorrect responses for the easy instances.*

Going back to fig. 6.4, we can revisit the summarised indicators of the three families. Looking at the two main clusters and the worse results of the shaped-up models on errors and difficulty concordance, we may rush to conclude that all kinds of scaling up and shaping up are inappropriate for ensuring user-driven reliability in the future. However, these effects may well be the result of the specific aspirations for these models: higher correctness rates (to excel in the benchmarks by getting more instances right but not necessarily all the easy ones) and higher instructability (to look diligent by saying something meaningful at the cost of being wrong). For instance, in scaling up there is a tendency to include larger training corpora (Villalobos et al., 2022) with more difficult examples, or giving more weight to authoritative sources, which may include more sophisticated examples (Schaul et al., 2023), dominating the loss over more straightforward examples. Also, shaping up has usually penalised answers that hedge or look uncertain (OpenAI, 2023c).

In summary, raw performance has increased, as has prompt stability, but so has the need for predictable performance due to the disappearance of adequate subject self-assessment. The disappearance of self-assessment has, although difficulty concordance is high, not been compensated with areas of perfect predictability, even at low difficulty. Good user-oriented performance predictability remains an issue.

Concrete Limitations

Finally, we must identify the limitations of our analysis. The first limitation being the recruitment of humans who are mostly non-experts. We have to take this into account when interpreting the calibrated difficulty values, which are usually high for some benchmarks, as there is a high number of questions that cannot be solved by the standard population. However, our motivation was to capture the same proxy of human difficulty for all datasets, also being more representative of the majority of users. A second limitation is that our sample of 'natural' prompts was collected from a diversity of sources, but we did not have access to the frequency

in which a prompt may appear in a real scenario. Lastly, we have only covered a sample of families. The GPT family has been at the forefront in performance and use over a few years, making OpenAI extremely influential in the development of other language models (J. Yang et al., 2024; W. X. Zhao et al., 2023). In fact, the OpenAI API has the most dependencies when the ecosystems of foundation models are analysed (Bommasani et al., 2023). LLaMA and BLOOM have a more open and systematic lineup of models, not only allowing for the disentanglement between scaling and shaping, but also paving the way for an incremental analysis of their evolution using our methodology and code, in a fast-changing context of LLM development. Highlighting the predictability issues of these families and introducing new abstractions and tools for analysis is then of utmost importance, enabling other players to explore different pathways for the scaled-up shaped-up models of the future.

6.5 Methods

We now explain our choices of benchmarks, prompt templates, difficulty functions, response scoring, general experimental design and the key metrics used to evaluate the models.

Benchmarks and factors of difficulty

For the generality of our analysis, we selected five very distinct benchmarks to reduce confounding factors as much as possible: simple numeracy (addition), vocabulary reshuffle (anagram), geographical knowledge (locality), elementary and advanced science questions (science) and information-centric transformations (transforms). These represent core skills (numerical, linguistic and knowledge) and more diverse ecologically-valid scenarios, with some of them having extremely simple formulations whereas some others requiring deep understanding of the information presented, as well as the integration of data from multiple sources. Closed-ended questions are typical of LLM research (OpenAI, 2023c), such as those found in the science benchmark, but gradually more open-ended tasks (addition, anagram, locality and transforms) better represent a wider and more realistic use of LLMs.

1. addition. This benchmark involves sums, prompting the LLMs by asking for the result of adding two addends (e.g., '`3 + 7 =`'). The examples in our analysis range from one-digit to one-hundred-digit additions. The way language models can not only memorise small additions but generalise to cope with any combination of larger digits makes this task very appropriate

to analyse difficulty trends. With respect to difficulty of addition, the number of digits and carry operations affect human performance on addition tasks.

2. anagram. The use of anagrams as a way of assessing aspects of problem solving dates back to 1916 (D. M. Johnson, 1966), and researchers have been using anagrams to examine a variety of phenomena, such as the cognitive processes involved in problem solving (Witte et al., 2002b). An anagram task is a word puzzle in which the subject is presented with a jumbled string of letters, and the objective is to find a word that can be formed using all the letters given. The examples in our analysis range from three-letter words to twenty-letter words. This task involves letter manipulation and good recall from an extensive vocabulary. One peculiar element of this task is that it is easy to verify. The difficulty of anagrams is mostly influenced by the frequency of the letters and the word, the number of letters, and the degree of rearrangement required.

3. locality. This benchmark contains questions relating to geographical knowledge, inspired by some cognitive models of distance estimation (Thorndyke, 1979). The examples in our analysis ask questions about the location and size of cities in relation to each other, by giving an input city and a randomly generated distance (d, ranging from 1 to 1000 km). The LLM is asked to identify the most populous city (the target city) within a radius of d km from the input city. This task requires geographical knowledge and reasoning. For this benchmark, potential human difficulty factors could be the city or country popularity, their population, etc.

4. science. This benchmark integrates multiple-choice questions from elementary science as collected by OpenBookQA, complemented with more advanced science questions from GPQA. They represent tasks that LLMs are likely to encounter in educational, academic and research settings (Birhane et al., 2023; Boiko et al., 2023b; Kasneci et al., 2023), some of them requiring several minutes to solve. The included questions are Google-proof (Rein et al., 2023). The science benchmark thus includes questions of varying levels of difficulty, as determined by human judgement, providing a lens through which to examine their handling of complex, data-rich tasks in specific domains.

5. transforms. This benchmark includes a comprehensive set of information-centric transformation tasks based on real-world scenarios. It focuses on domains that are most prevalent in the use of LLMs today (Zheng et al., 2024), while ensuring that there is a ground truth for evaluation. We integrate many data formatting tasks —a well-studied area in LLMs (Jaimovitch-

Lopez et al., 2021)— but also new tasks about world knowledge, information retrieval, advertising, administration, coding, scheduling and retailing. The outputs for transforms may require extensive elaboration of the input (hundreds of characters) to form a correct answer, which can also be hundreds of characters long. The aim was to simulate as closely as possible the complexity and depth of real-world questions in a controlled experimental setting. For task difficulty, given the heterogeneity, the main factors are as general as character and word counts, and the Levenshtein distance between input and output as a proxy of transformation effort.

For the previous domains we can find intuitive human difficulty proxies, some developed in the literature. See appendix C.4 for further details on the definition of difficulty metrics and the abilities behind the features used for their definition. Using the results from human study S1, we select the difficulty functions that are most correlated with human expectations (table C.4): f_{cry} for addition, f_{let} for anagram, f_{pop} for locality and $f_{\text{w+l}}$ for transforms. For science, we blend and calibrate the two original human metrics into one f_{hum}. For all benchmarks, we map the original difficulty functions using a logistic mapping into a scale between 0 and 100 that corresponds to the probability of human failure as estimated by humans themselves. We need to take into account that these values are an estimate (from the human sample in S1, about their expectations, and fitted with a two-parameter logistic function), so these values between 0% and 100% have to be taken with care, especially with small differences. See appendix C.8 for details. Nevertheless, having all difficulties in the same human-expectations scale helps with the comparison of the benchmarks.

Data and Prompt Collection and Generation

We first describe how the examples were collected or generated, and then the 15 prompt templates that were used with each them.

1. addition. We randomly generate 5000 instances, where each addend is sampled uniformly from 1 to 100 digits. We then remove those instances whose $f_{\text{hrm}} > 50$ to prevent instances with similar or identical number of digits in both addends from dominating the upper difficulty bins. This is because, for example, if the difficulty is the harmonic mean, the bins with $f_{\text{hrm}} > 90$ would be dominated by instances whose addends both have very high numbers of digits (i.e., at least 82 digits). A similar phenomenon also occurs with other difficulties, but with the previous criterion considered, the problem is well mitigated. This results in a final sample of 3142 instances.

2. anagram. We use Google Web Trillion Word Corpus (Franz & Brants, 2006), containing the frequency of over 300K most commonly used single words on the Web in English. With this corpus, we randomly choose up to 100 English words with 3 to 20 letters, resulting in a total of 1570 words; there are fewer than 1700 instances because there are fewer than 100 English words with 17 to 20 letters. Then, we shuffle the order of letters randomly to map these words into 1570 anagrams. We make sure the resultant permutation does not equal the original word.

3. locality. We use the World Cities Database (SimpleMaps, 2023), which provides an up-to-date database of the world's cities and towns. With this database, we first exclude cities with non-unique names across the globe. Next, we remove cities with more than one word or non-standard 26 alphabetical letters (e.g., Buenos Aires or Ch'ŏngjin) to enhance the quality and ease of the response scoring method. After the previous selection procedure, we seek to form a final sample that covers instances with different difficulty levels (or bins) as equally as possible. Thus, we perform binning on the difficulty function (f_{pop}) into 100 bins in which we extract up to 50 instances from each bin randomly, resulting in a total of 2341 instances; again, there are less than 4500 instances because some bins contain less than 50 instances.

4. science. This benchmark is built by integrating multiple-choice questions from educational settings: OpenBookQA (Mihaylov et al., 2018) and GPQA (Rein et al., 2023). OpenBookQA is a collection of multiple-choice questions in elementary science, based on 1329 established facts. We sampled 1000 questions randomly from OpenBookQA. To complement the benchmark with more advanced science questions, some of them requiring several minutes to solve, we included GPQA (Rein et al., 2023), a dataset containing 546 graduate-level questions written by domain experts that challenge LLMs to demonstrate a deep understanding of biology, physics and chemistry. We exclude two lengthy questions that exceed the context window limit for some of the models that we analyse.

5. transforms. This benchmark includes a comprehensive set of information-centric transformation tasks based on real-world scenarios. We integrate those data formatting questions from (Jaimovitch-Lopez et al., 2021), a data wrangling dataset, we take some from (Mishra et al., 2022), a 'natural instructions' dataset, although mostly manually regenerated, and we also introduce new tasks about world knowledge, information retrieval, advertising, administration, coding, scheduling and retailing, reflecting a very

wide range of real user interactions with language models. The benchmark integrates 73 different tasks, with ten instances each, totally 730 items.

Noteworthy, addition, anagram, locality and parts of transforms are newly introduced in this work. All five benchmarks are additionally supplemented with human data (see appendix C.5) for calibrating difficulties and supervision, as well as a new variable describing the human-calibrated difficulty for each data instance.

Each example in each benchmark will be run through a LLM using 15 different prompts, which are the same for all examples in the benchmark. The generation of prompt templates aims to fulfil three desiderata. First, the prompts should be as natural as possible, as this tries to model a situation where humans would interact with LLMs as they would ask other humans. Second, these prompts should be derived from or inspired by real-world sources, except for minor variations and adaptations. Third, we need to have sufficient coverage for and diversity of prompt templates, to robustly analyse sensitivity, omitting those that are too similar. This process results in 15 natural prompt templates for each benchmark, extracted or inspired from textbooks, scientific literature, academic exams and the Internet. Appendix C.2 describes further details about these prompt templates and their sources.

Response Scoring

Scoring the validity of the responses of LLMs can be challenging, given that their raw text response can vary in very different ways. For example, some responses are very elaborate, while some other responses are concise and straight to the point. Some responses are unrelated or digress from the proposed question, or are just excessively verbose, providing the answer within a larger response sequence surrounded by arbitrary information. Because our analysis uses three classes: correct, incorrect and avoidant, the confusion matrices have nine cells, making grading more challenging, and the traditional intuition and terminology of false positives, false negatives, sensitivity, specificity, precision and recall cannot be extended easily to these 3-outcome situations. In appendix C.11 we discuss how different groups of cells are named.

Manual scoring becomes infeasible due to the massive amount of answers we collect (approximately 4.2 million). Fortunately, despite the models' arbitrary responses, they do exhibit a set of common patterns. We succeeded in scoring these responses satisfactorily using simple algorithmic conditions and regular expressions that provide great scoring accuracy (detailed in appendix C.3).

Experimental Setup

The LLMs are described in table 6.1. All models were queried with the temperature parameter set to zero and no system prompt. For local inference we made use of a shared cluster of six nodes with 8 × NVIDIA A40 48GB GPUs. All local inference was single node, made use of the Hugging Face Transformers and Accelerate libraries, and was without quantisation of the models, with the exception of BLOOMz (see below). Total compute estimate for all experiments (including reruns and discarded results) is estimated to be about 100 compute days on a single 8×A40 node.

- **GPT**: We used ten models from OpenAI's Generative Pre-Training (GPT) family (Radford et al., 2018). The first four models, GPT-3 Ada, Babbage, Curie and Davinci, are the original 'raw' models in the family (Brown et al., 2020). The subsequent three are the later and more powerful models variants, known as the InstructGPT versions of Davinci: text-davinci-001, text-davinci-002, and text-davinci-003 (Ouyang et al., 2022), which are shaped up by fine-tuning with human feedback. The last three models are also fine-tuned with human feedback but additionally including a moderation post-filtering mechanism (OpenAI, 2023c). GPT-3.5-Turbo was build 'gpt-3.5-0301' (March 2023), and the two GPT-4 models differ in the time of their build ('gpt-4-0314' and 'gpt-4-0613'). These models were all accessed through the public API. We used the ChatCompletion API (https://platform.openai.com/docs/api-reference/chat/streaming).

- **LLaMA**: We used the four different scales of the first LLaMA version (Touvron, Lavril, et al., 2023): 7b, 13b, 30b, and 65b. For LLaMA-2 (Touvron, Martin, et al., 2023), there is no 30b variant available, but we used all other sizes (7b, 13b, 70b) including the corresponding chat variants, which incorporate various shaping techniques. All inference was run locally, except for LLaMA-65b, for which we used the Hugging Face API, and LLaMA-2 (non-chat) for which we used the Together.AI API.

- **BLOOM**: We used the six different scales (560m to 176b) of the BLOOM (BigScience et al., 2023) and BLOOMz (Muennighoff et al., 2023) models, the latter of which was an update that added (multilingual) multitask fine-tuning (also known as instruction tuning). As before, all inference on the small models was run locally. The biggest variant for BLOOM was run through the Hugging Face API. BLOOMz was run locally, but with NF4 quantisation (Dettmers et al., 2023) to fit into a single node.

The number of tokens was adjusted for the benchmark: addition =256, anagram =72, locality =132, science-OBQA=72, science-GPQA=384 for all models except for GPT-3.5 and GPT-4 where we used 1000 tokens. For transforms, we used the formula round(max(72,output_length))*3/4. All these numbers ensured that we could get long enough responses that include the answers for ~99% of instances while substantially reducing the cost. We use the default values for the stopping condition and the rest of parameters.

Evaluation of Models

For each difficulty function, we rank the data examples and separate them into 30 same-sized bins based on their difficulty values. With this, we calculate bin-wise correctness, incorrectness and avoidance rates. Then we plot these rates with the stacked bar chart (fig. 6.1), with which we calculate Spearman rank correlation (table C.7). Similarly, we illustrate prompt sensitivity of correctness, incorrectness and avoidance by plotting the performance of each individual prompt template for these dimensions across each model (figs. C.16 to C.18).

In addition, we delineate six reliability indicators for all the models in OpenAI's GPT, Meta's LLaMA and BigScience BLOOM families, as depicted in fig. 6.4. There are three categories of answers: correct ($\mathbf{c}$), avoidant ($\mathbf{a}$) and incorrect ($\mathbf{i}$). By separating correct against avoidant or incorrect ($\mathbf{c}$ vs $\mathbf{a} + \mathbf{i}$), the design or evaluation focus is put on *accuracy*, whatever damage the errors may do, but if correct or avoidant is placed against incorrect ($\mathbf{c} + \mathbf{a}$ vs $\mathbf{i}$), the focus is put on *reliability*. Instead of non-incorrect, we use the term *prudent* to refer to the group of correct or avoidant answers as a whole. Taking into account these groups we have two versions for each of the following indicators:

- **Proportion**: this measures the percentage of some of the groups of responses. In particular, the **correctness proportion** is the probability of giving a correct answer, i.e., $\mathbb{P}(\mathbf{c}\langle j, p\rangle)$, where j and p refer to an instance and a prompt for that instance respectively, and $\mathbf{c}$ represents correctness. Also, the **prudence proportion** is the probability of giving a prudent (non-incorrect) answer, i.e., $\mathbb{P}(\neg\mathbf{i}\langle j, p\rangle)$, where $\mathbf{i}$ represents incorrectness.
- **Prompting Stability**: this is the probability that the answer to an instance remains in the same group after changing the prompt. Let us define $s^{\mathbf{c}}$ as $\mathbb{P}(\mathbf{c}\langle j, p'\rangle \mid \mathbf{c}\langle j, p\rangle)$, where j refers to an instance, and p and p' refer to two prompts for that instance (not necessarily different). This just measures the probability that given a pair of instance and prompt that is correct (sampling uniformly from all these positive pairs), we still get a correct answer if we

sample another prompt. Similarly, we define $s^{\neg\mathbf{c}}$ as $\mathbb{P}(\neg\mathbf{c}\langle j, p'\rangle \mid \neg\mathbf{c}\langle j, p\rangle)$. Finally, we define **correctness prompting stability** as $s_c = 0.5 \cdot (s^{\mathbf{c}} + s^{\neg\mathbf{c}})$ and **prudence prompting stability** as $s_p = 0.5 \cdot (s^{\mathbf{i}} + s^{\neg\mathbf{i}})$. It can be shown that these metrics go between 0.5 and 1; we scale them to go from 0 to 100.

- **Difficulty Concordance**: this measures the degree to which higher difficulty implies lower quality of results. We will use the generality metric introduced in (Hernández-Orallo et al., 2021), as it aligns precisely with the concept of difficulty concordance. Technically, generality is a non-parametric metric that measures how much the mass of success conforms to a step function. If success were distributed like a descending logistic curve, generality would be equal to the maximum slope of a descending curve, i.e., the steeper the slope, the higher the generality metric gets, and thus a higher level of difficulty concordance. A model being good for all instances up to a given difficulty and then bad for more difficult instances would have perfect concordance. This is thus not the same as correlation, as shown in table C.7. Again, we define two versions, **correctness difficulty concordance**, which calculates the generality for the correct answers, and **prudence difficulty concordance**, which calculates the generality for the prudent (non-incorrect) answers. We transform it with $x/(x+1) * 100$ to get a value between 0 and 100. For science we discount 25% of non-avoidant response to account for random guess.

We suggest the use of these six reliability metrics as a first summarised inspection of the reliability of any existing or future LLM. In fig. 6.4, we do this by averaging the values procured from the five benchmarks to provide a succinct summary of the reliability fluctuations of the three families (detailed data in C.11).

Following the advice in (Burnell, Schellaert, et al., 2023), we strongly recommend that these metrics are always accompanied by a detailed analysis and breakdown of results, as we have done in this paper with the other plots.

Chapter 7

Conclusion

Summary and Contributions

Chapters 1 & 2

In the introduction, we laid out a motivation for focusing on prediction as a way to improve the practice and science of evaluation: prediction is core to the philosophy of science, it is core to evaluation in other fields such as animal cognition and psychometrics, and it is the central language of the field of machine learning itself.

Prediction is especially pertinent, as laid out in our background section, because prediction holds such a prominent role in the current challenges of evaluation. Both for traditional but still relevant issues, such as distribution shift, the refinement problem, and a multitude of conflicting goals, as well as for more modern stressors induced by the advent of general purpose AI, such as the loss of a prediction target, an increase in scope, and notable contamination issues, we find that they are all significantly related to prediction.

Chapters 3 & 4

In Chapter 3 we formulated an intuitive formal framework, considering evaluation as a learning algorithm that produces a score predicting model from empirical

data. We discussed the inevitable consequence of needing to meta-evaluate evaluation methodology, and clarified the challenges involved. This included heavy data requirements, dependency on subject and task-specific data, as well as the need to balance score predicting power and subject predictability on one hand with subject performance on the other.

This framework then helped us reframe the challenges of evaluation in the language of statistics and machine learning, providing a solution to the refinement problem by conditioning on the input variable X, which also partially addresses the distribution shift problem. For challenges that remain unsolved. e.g. out-of-distribution prediction and the large scope of evaluating GPAI systems, clear analogies to machine learning –i.e. the training rather than evaluation– can be drawn, providing a way forward through scaling up data, focusing on generalisation, or introducing meta-learning.

Our framework also helped us identify more than fifteen applications and over twenty types of methods, which we organised by defining several keystone properties of score predictors: the kinds of generalisation; the level of information used about the subject, instance, and their interaction; the levels of anticipation; finishing with various properties of the prediction target, including e.g. continuity, direction, or proportionality.

Many of the methods and applications were previously unconnected, and this synthesis has proven a bountiful source of ideas yet to be explored to their full potential. Examples include gaps in the literature such as instance-based scaling laws; next-gen methods like assessor models; and the surprising parallels between unrelated fields, e.g. recommender systems moving from inner to full generalisation to maximise data value, which parallels the road ahead for evaluation. The Future Work section below presents many more.

Chapters 5 & 6

Chapter 5 marked the start of our empirical investigations. We focused on assessor models: the natural implementation of our formalism. We conducted a varied series of experiments, fine-tuning DeBERTa models with evaluation records from the BIG-bench suite of benchmarks, to become score predictors for large language models with up to 128 billion parameters. We demonstrated our assessor method, achieving AUROC scores of over 0.85 for some tasks. More importantly, we identified several factors that influence score predicting power that might carry over to different methods, including the presence of multi-task and multi-subject data, both of which can –though not for every task– significantly help improve score predicting capabilities.

In our second empirical study, we conduct an extensive human-centred study of language model performance. Starting from human-derived notions of difficulty, we analyse performance, question avoidance, prompt sensitivity, and human supervision across five different tasks over varying difficulty levels. We also add two other dimension: that of scaling language models, i.e. making them bigger in terms of parameters and ingested data, and that of shaping them, making them more instructable and easier to use. Over more than thirty models from three different language model families (GPT, Llama, and BLOOM), we find that while human difficulty correlates well with LM performance, there is too little of a step shape in the difficulty-performance plot to facilitate confident prediction. There is no region –regardless of difficulty– where performance is perfect, and newer subjects improve by gaining ground on the medium and hard instances, rather than the easy ones.

We find that the problem is also exacerbated by, although decreasing over time, ever-present prompt sensitivity. This prompt sensitivity puts an upper bound on predictive accuracy for human-sourced score prediction: we cannot expect humans to model the performance differences on semantically identical questions, especially as the relation between format and score is utterly chaotic. Improvement must thus come from the subject, not from the score predictor. Lastly, we also confirm that a lack of predictability is indeed a problem. One, it is not compensated for by a subject's self-assessment. Quite the opposite, self-assessment worsens as models increase in scaling up and shaping up, leading to *more* incorrect answers with newer models. Secondly, neither is it compensated for by human supervision, with users over-relying on their faulty expectations of performance and failing to properly verify subject responses.

—

Even with much of Chapters 3 and 4 left to publish, our work so far has led to 11 published research papers (see appendix A), the two most extensive ones receiving their own chapter (Chapters 5 and 6), the others woven throughout to form a paced and structured story.

Our last chapter on human-sourced score prediction is a good confirmation that our investigation in predictability should be the beginning of a larger research effort, not just a curiosity. The problem of predictability is real, it is tangible, and it is not getting better! As a start, in Burnell, Schellaert, et al. (2023), we therefore called upon the community to report more granular evaluation results; so that we would have a better chance of connecting instance-features, subject-features and scores, and if not possible, to make instance-level subject predictions available, so others can take up the task. That call has motivated or been echoed

in various efforts throughout the community, from policy and safety (Anwar et al., 2024; Díaz-Rodríguez et al., 2023), over philosophy (Browning & LeCun, 2023), to concrete evaluation efforts (Balog & Zhai, 2023; Collins et al., 2023; Salewski et al., 2023).

Reflection on Research Objectives

Objective 1: Develop a formal framework

> *First, we aim to develop a formal framework that supports the analysis of evaluation as a prediction problem. It should allow expressing the problems of evaluation, e.g. distribution shift and the refinement problem.*

We propose to interpret evaluation procedures as learning algorithms that produce score predicting models based on subject and instance info. We copy the formalisms of machine learning, restricting it only in input and target space. Our interpretation is simple, but nonetheless constitutes a framework that is a rich source of insights and parallels.

This has allowed us to copy concepts such as generalisation and empirical risk, and map them onto evaluation techniques to pinpoint problems that would be obvious in a machine learning setting. For example, copying specifically from section 3.5, making refined score predictions for a specific question requires a predictive model that is conditional on the random variable that materialises that question, the same way a machine learning algorithm conditions on its input variable. Evaluation accuracy degrades with distribution shift, because any predictive model degrades with distribution shift, and predicting out-of-distribution is hard in general, see e.g. our experiment in section 5.3 for example. A last example is our definition of contamination: those instances where a good score predictor expects bad performance, but the subjects nonetheless perform well.

The challenges of evaluation can clearly be expressed in our framework.

Objective 2: Survey and integrate the literature

> *With an adequate formalism in hand, we must survey, integrate, and organise the techniques and applications of evaluation. We must find overarching properties and use those create taxonomies, draw new connections, and synthesise the science of evaluation.*

Chapter 4 is a testament to the completion of this goal, connecting a wide range of applications and methods through the unifying lens of our framework and condensing research efforts spanning hundreds of papers to forty-ish pages and a few summary figures.

Especially the generalisation properties –inner, subject, instance, and full– have proven fruitful for structuring the various predictive evaluation methods, explaining for example why particular methods are used for architecture search as a function of whether they generalise to new subjects or not. We have also seen the close parallels with recommender systems literature: the field has moved from inner generalisers such as matrix factorisation (section 4.2.2.1) to assessor-like techniques such as Two Tower methods (section 4.2.5.3).

Another example is how assessor models, which are the natural implementation of our framework, have been discovered a few times independently in literature for separate use cases, e.g. Shnitzer et al. (2023) for routing and NannyML (2024) for performance monitoring, or at the aggregate level for scaling analysis as in Q. Zhang et al. (2024).

Objective 3: Improve a concrete application of evaluation

> *The synthesis of the previous objective should lead to the identification of gaps in the literature and new avenues of improvement making evaluation more predictive. Those improvements or ideas should be demonstrated and empirically validated for at least one concrete practical application of predictive evaluation.*

Completion of this objective is more mixed. On one hand, the successful synthesis created more ideas and connections that could realistically be explored. For example, while we proposed and motivated instance-based scaling laws with multi-dimensional subject features (section 4.2.3.2), we did not commit to an experimental validation. The objective foresaw this and bode us to validate at least one idea empirically.

We did this in part by demonstrating assessor models in Chapter 5. However, this was not a concrete practical application or use case, and instead this chapter focused on more general investigations that could affect several score-predictors and their use cases simultaneously, especially assessor-based instance-generalisers like (Shnitzer et al., 2023) for model routing or NannyML (2024) for monitoring. Nonetheless, our insights beg for further practical demonstration.

With respect to human score predictors, we have found human derived difficulty to correlate well with performance, but also found the difficulty-performance plot shows too little of a step shape for difficulty to act as a really accurate predictor. Important to note however, is that our investigation of prompt sensitivity simultaneously also highlighted an upper limit for the accuracy of human instance-level score prediction. We cannot reasonably expect humans to model these chaotic and alien performance fluctuations, and future work should not ignore improving the subject's predictability to start with.

This concludes a composite result with regards to the objective, with a lack of demonstrated improvement for concrete use-cases, but with various insights that should affect them nonetheless.

Objective 4: Produce fundamental insights

> *The same formalism and synthesis should also lead to deep insight into remediating at least one of the fundamental challenges in evaluation, e.g. distribution shift, the refinement problem, or contamination. While this is an abstract objective, success can be measured in the plausibility of uncovered research directions, or if practicalities allow, implementation and demonstration of new paradigms of evaluation.*

Note: this discussion will be largely based on section 3.5.

We think there are at least five findings that match the requirements of the objective.

(i) Barring other difficulties such as distribution shift, refined score prediction is no longer a problem due to machine learning techniques such as assessors. This is at least true in theory, since practical constraints such as data requirements might still prohibit accurate fine-grained prediction. (ii) On the other hand, solving the refinement problem for human score prediction requires improving subject predictability first. Prompt sensitivity acts as an upper-bound on what is feasible for users to achieve in terms of accuracy. (iii) If subject predictability does not improve, the increased scope of GPAI evaluation will require scaling up evaluation

data and improving generalisation of score predictors. (iv) As with refinement, observing the instances in a conditional score predicting model $s(r|x)$ can help with certain types of distribution shift. (v) If observing x is not possible, we need score predictors that can handle high-level descriptions of distributions and instances, e.g. task demands as in capability oriented evaluation (section 4.2.5.4), which do not require direct sampling from an inaccessible data generating process.

—

While objective 3 has not been met in full, all other objectives –1, 2 and 4– have been completed successfully.

Reflection on Research Hypothesis

First, let us restate our research hypothesis:

> *By formally framing evaluation as a prediction problem, we can (i) bring together disparate methods and applications of evaluation of artificial intelligence, (ii) increase evaluation's predictive power for concrete applications that require it, and (iii) remediate debilitating issues in the science of evaluation, both old and new.*

The answer to the first question is a resounding yes, as demonstrated by the richness and variety of Chapter 4.

Regarding the second question on concrete applications, our own evidence is insufficient to make strong claims. However, papers from other authors (e.g. C. Yang et al. (2019) and Q. Zhang et al. (2024)) have demonstrated noteworthy improvements in score predicting capacity for concrete use cases by borrowing techniques from other various subfields of AI, e.g. meta-learning or recommender systems. Although these papers do not explicitly refer to the notion of evaluation as prediction, their results nonetheless confirm our hypothesis, even if more by accident.

Regarding question three on remediating debilitating issues, with assessors we have made non-trivial headway in dealing with the refinement problem. Although we had not yet given convincing demonstrations, others have (Hu et al., 2024). The idea is flexible and execution is currently primitive, so there should be room for further advancement. Similarly, dealing with certain types of distribution shift is also possible by conditioning the score predictor on the instance-level inputs, as exemplified in our example in section 3.1.6.

To further put our money where our mouth is, we can adopt Whewell's (1849/ 1969) viewpoint that served as motivation for this dissertation, i.e. that theories are confirmed by their prediction of surprising new results, and apply it to our own theories. Personally, we were particularly surprised by the strong parallels between recommender systems and evaluation. Notably the two-element input space –users/products versus subjects/instances– and how it affects the different types of generalisation: both in evaluation and recommender systems we see inner, user/subject, instance/product, and full generalisation. Even the mathematical foundations align, e.g. matrix factorisation (section 4.2.2.1), which is similar to how factor analysis based evaluation is done in psychometrics.

We therefore conclude hypothesis (i) as confidently confirmed. And although the confirmation of hypothesis (ii) and (iii) rely on external work that has not been influenced by our framework, our framework did lead to the same insights. For this reason we also conclude hypotheses (ii) and (iii) as confirmed.

Reflections on Research Methods

Any reflection on our research methods must be dual, as there are both the theoretical and literature-based methods from Chapters 3 and 4 and the empirical, experiment-driven methods in Chapters 5 and 6.

With regards to the former, it should mainly be noted that our literature review, although varied and extensive, was not methodological. It is not unlikely that we have missed interesting approaches among the diverse sub-fields of AI, or science in general. Nonetheless, a fully methodological review would likely have been more mechanistic and reductive, and while it could have supplemented the current effort, it could not have replaced entirely the organic and network-driven collection of research.

With regards to the latter, i.e. the experimental investigations, Chapter 5 suffers heavily from lack of statistical certainty and clear conclusions. Due to high number of varied research questions, both data and experimental setting were not necessarily optimised for answering each question individually and thoroughly, instead producing weak signals across the board. However, this explorative nature was in part also the aim of the work, as it provides groundwork for setting prioritisation and creating dedicated experiments for several interesting hypothesis with preliminary support.

For Chapter 6, internal validity has improved, although the lack of a solid strategy for aggregating results prevents assigning high confidence to several conclusions,

as even with eye-ball tests not all patterns can be clearly observed. With regards to the experiments dealing with human supervision of LLMs, external validity is lacking, as the rather sterile setup of human-computer interaction is not ecologically valid. However, proper HCI experiments are typically orders of magnitudes more expensive, and our results –focused on collecting large amounts of data over several datasets and models–, are a necessary first step in identifying the models and datasets that are most relevant for confirmatory experiments.

Personal Reflections

A lot of the content in Chapters 2 to 4 is newly written for the dissertation. Of course we had thought deeply about these things before, but it is the first time that all of our thoughts have been collected and structured, with inconsistencies and open questions needing to be resolved as much as possible. On the other hand, the first experiments for Chapters 5 and 6 are dated spring 2022 and summer of 2022 respectively, more than 2 years ago at the time of writing. This might have been the wrong order of doing things. Figure 4.8 is entirely new, but has been immensely fruitful already in the writing of this dissertation. Connecting the levels of anticipation (section 4.3.2) to the various applications would have helped us identify concrete use-cases and applications to demonstrate assessors in earlier. As Hernández-Orallo et al. (2022) got accepted early in the doctoral studies, instead of double checking our assumptions, integrating literature more firmly and enjoying the reduced pressure of publication, we went on to try and publish more, and with a focus on empirical studies, while there was so much left to plainly think about. Of course it is difficult to organise so many disparate methods barely one year into the PhD, but there is a reason students are usually recommended to start with a literature review.

We have also recently started the design of a community-oriented benchmark focusing on score predictors and subject predictability, in order to design metrics and incentives that accommodate the trade-offs between performance and predictability, closely related to the discussion in section 3.3. This also we should have done earlier rather then later. Thinking about how to evaluate evaluation while having control over both subject and evaluation, and with participants with different incentives, has forced us to think more clearly about the benefits of score prediction, with less excuse for generalisations as in Hernández-Orallo et al. (2022) or tailoring to ecologically uninteresting niches (L. Zhou et al., 2022). It would have created more community participation, and made the results of Chapter 5 more immediately applicable than they are now. In our defence, the field of AI

has changed tremendously over recent years, and maybe this idea made less sense back then.

There is no regret however with regards to the choice of evaluation as primary topic of study, nor with following the hunch that performance prediction is central to understanding evaluation as well as artificial intelligence. With the field moving so fast it is easy to miss the forest for the trees; to reinvent the wheel; to prod the algorithmic alchemy in darkness, ultimately missing the bigger picture. In a way, the problems of AI evaluation are the same as they were five or ten years ago, and the fact that we can make progress in AI systems despite these problems existing does not invalidate them being an actual problem. Quite the opposite, the advent of GPAI has exacerbated old challenges, pushing minor problems such as contamination into the limelight, to the point were results are barely trustworthy: practitioners routinely take 10-15% margin on reported results. Engineering excels at making pragmatic advances, at realising concrete benefits today even under intense epistemological uncertainty, but eventually it must wait for science to catch up. We have enjoyed trying to catch up.

Limitations

First, some concrete problems. As noted already, Chapter 5 is plagued by variability, and thus statistical insignificance. The results are therefore too unreliable to draw confident conclusions, and are mostly useful as indications and priors for future research. The work could have benefited from fewer, but more robust experiments. In Chapter 6, a limitation not yet mentioned is that the used difficulty ranges might not be ecologically valid, in that users might realistically only have needs for difficulty bins 1 to 5, rather than 1 to 30. The trends we have seen may be less strong in this reduced range.

We have not paid any attention to the philosophy or mathematics of measurement (Tal, 2020), but measurement is a pre-requisite for both prediction and learning. We have assumed it is possible to measure the quality of a response, and to measure whether a prediction holds. Potentially, integrating measurement as first-class citizen would lead to additional insight. Similarly, we have not explicitly explored any connections to complexity science or cybernetics, which often also model subjects as black boxes, observing only inputs and behaviour.

Although appendix B.1 and appendix B.2 indicate some sort of connection between predicting scores and solving the original problem, we did not substantially explore this. Especially in sequential decision problems, we should expect a strong connection between estimating the score a subject will achieve, and assessing the

value of a certain state, which is exactly the problem sequential decision problems try to solve. The estimation of value/score is also for a two-dimensional input space: (subject,instance) vs (action,state). Relatedly, we have consistently considered grading –deciding the score when behaviour is observed– to be out of scope. But for plenty of single-step instance-level use cases a post-hoc estimation of scores is sufficient, unless we care about costs and resources in particular. Even with multi-task sequential decision problems, the subject could present an execution plan or policy to take into account for score estimation, removing the necessity to fully anticipate behaviour. Note again the symmetry: if a plan is conditional on future states (e.g. because we are unsure about environment dynamics), we tend to call it a 'policy', which is a term also used to refer to the subject sequential decision problem, and which is the topic of evaluation in the first place.

It has become clear that the representation space, i.e. how to describe subjects and instances, either in aggregate or not, holds a lot of importance for building flexible and generalising score predictors. In this regard, we have focused maybe too much on methods of inference, rather than the problem of translating human models of the data distribution into machine interpretable representations.

We have neglected methods for score prediction focused on sequential decision problems and reinforcement learning. Only briefly mentioning shielding in section 4.1.3.4 and section 4.2.6. Given its purported relevance to anticipative score prediction, the corresponding methods deserve more attention.

We have also neglected the improvement of human understanding of artificial intelligence as an application for evaluation, i.e. evaluation for explanation, even though it was one of our principal motivations. Not all predictions entail explanations, and with our focus on complex score predictors, we have not dealt with simple models that humans could interpret. However, as Breiman (2001) argued, "the soundest path is to go for predictive accuracy first, then try to understand why", and investigations, and efforts like (Q. Zhang et al., 2024), applying Shapley feature importance analysis to subject features, are promising. But while we evaluate predictive accuracy of assessors in Chapter 5, we do not quite investigate the predictive powers of the other methods in section 4.2 in an absolute sense. The papers at hand usually do, but improvement over some baseline is enough to be interesting for the respective use case. To go from prediction to understanding, we should investigate what level of predictive power is needed to gain concrete insights. E.g. is an AUROC of 0.7 sufficient, or do patterns only become clear or meaningful at AUROCs of 0.9 or higher? As Shmueli (2010) says, "predictive modelling is nearly absent in many scientific fields as a tool for developing theory". Maybe that is for a reason?

Lastly, the relation between predictability and safety, i.e. how does a certain level of predictability improve concrete measures of safety, is one that we have not explored in detail, although it was used as motivation for a variety of applications, especially for optimising inference (section 4.1.3). We have briefly discussed our proposed Perfect Predictability Region metric, but given the importance of safe AI, a more formal and comprehensive analysis would have been welcome.

Ethical Considerations

The ethical implications of this work are limited. We do however want to explicitly and pre-emptively warn against possible misinterpretation of our results that could lead to misuse.

Foremost, it should be clear that any prediction, including the prediction of scores, can be wrong; if score predictors would be used in domains such as policy enforcement or auditing of AI systems, care should be taken as to not infringe on civil and corporate rights, e.g. barring the deployment of AI systems due to a predicted lack of safety. In those scenarios, and especially when non-causal score predictors are employed, the identified subject weaknesses should only be used as directions for further testing and evaluation, such that objective measurements of safety can be made.

Similarly, the prediction of successful subject behaviour is not sufficient argument to shift the liability for the subject causing actual harms during execution. At all points it should be taken into account that a score predictor can be wrong, and that chance should be carefully balanced against possible risks.

Lastly, while we study the evaluation of artificial intelligence systems, many of the philosophical and theoretical principles could apply to the evaluation of humans as well. However, barring additional human-specific research, we can not assume this to be the case, and we do not support extrapolating any of our results to situations wherein human performance is predicted, e.g. in recruitment.

Future Work

Plausible Directions of Improvement for Machine Score Predictors

In the context of methods like assessor models, there are plenty of options to vary the experimental conditions explored in Chapter 5 and to correspondingly

increase performance. A major and obvious direction is the exploration of larger, text-to-text architectures, such as the latest generation of Llama models (Dubey et al., 2024). But also for example, exploring few-shot prompting for the assessor (rather than for the subject), where (subject, instance, score) triples could be included in the assessor prompt. We should also investigate why multi-task or multi-subject training helps for some tasks, but not for others. Perhaps the data ordering had a significant effect here, and it could help to finish the learning process with the specific task of interest.

Machine learning practice should be a rich source of improving evaluation in general, the organisation from Chapter 4 in mind. In particular, the symmetries between recommender systems and score prediction use cases are noteworthy. Both have dual input spaces, product/user for one, subject/instance for the other, and both have uses for inner generalisation, yet ultimately have to deal with generalisation across both axes. Papers like C. Yang et al. (2019) and Q. Zhang et al. (2024) already explicitly borrow from the recommender system literature, which is well developed, and is backed by substantial economic interests. Exploring methods like Two Tower Models (P.-S. Huang et al., 2013) could be worthwhile.

Another machine learning-inspired option could be to develop Mixture-of-Expert assessors, with heavy focus on specialisation in the experts, as in Dai et al. (2024). While maintaining predictive accuracy, these smaller experts are much more interpretable than larger and dense variants, which could prove useful for learning what makes instances hard and subjects capable. In a multi-subject setup, would experts specialise per subject? What would that tell us about the effect of multi-subject learning? It could also be possible to integrate a reject option directly in the router of an MoE subject, which would have the learning from the large collection of training instances.

Lastly, we could apply label noise methods (Frenay & Verleysen, 2014) to deal with multiple-choice questions and contamination, where a good score might not be representative, or we could adapt those methods attempting to tackle robustness to distribution shift and out-of-distribution samples, e.g. meta-learning (Grąbczewski, 2014; Hospedales et al., 2020; Vilalta & Drissi, 2002), or invariant risk minimisation (Arjovsky et al., 2020). In general, there remains much to explore with regard to adopting machine learning techniques for improving evaluation.

A more fundamental paradigm shift can be found in including more information about the subject's training data, which seems to us almost inevitable in order to create truly capable score predictors. It could help improve nearest neighbour-based assessors, as in Hu et al. (2024), or allow measuring the variability of

performance for a single instance across different epochs, as shown to be very informative by Swayamdipta et al. (2020). More prominently it seems necessary –or at least very helpful– to deal with contamination, as shown in the Datamodels approach (Ilyas et al., 2022). To deal with the enormous scope of evaluating GPAI system, it might also be more realistic to observe scores for training instances rather than to increase the test set to the same scale as the training set.

There is also a need for an adequate representation space wherein tasks and distributions can be expressed, without needing to sample concrete instances. Oftentimes, we know we want an AI system to tackle a particular task, but we do not have representative samples of the task available, e.g. because we expect some form of distribution shift. Humans should be able to describe their expectation of the task and data distribution, either in absolute terms or in differences from a reference set, in way that can serve as an input for a score predicting model. For this purpose, capability-oriented evaluation section 4.2.5.4 seems promising, with high-level task demands as instance-features being a flexible interface for communicating between man and machine.

And since we are talking about human-computer interaction, as described in section 3.4, including the user as a parametrised input for a score predicting model is a sensible but unexplored route to disentangle features that are informative for scoring, and would allow us to anticipate the validity of a system for a particular problem and a particular user.

Plausible Directions of Improvement for Subject Predictability

There is a limit to what score predictors can achieve without subject predictability improving. Unless there is a pattern to be found –and we have not found one yet–, performance variation between semantically equivalent questions is a good example of something that we cannot realistically expect score predictors to accommodate. Especially with respect to human score predicting, we are limited in the complexity of the performance model we can expect users to have. Educating users will of course help, but there is need for improvement in subject predictability itself.

We could also adapt the loss function used to train the subject to maximise difficulty concordance and to reduce incorrect-to-correct errors in human verification. Easy examples should be prioritised, and deceptive outputs that escape human supervision should be penalised, potentially in the same way reward models are trained right now with human feedback (Lambert et al., 2024). For this, collective efforts are needed to build larger datasets of human difficulty expectations

and output supervision. With this data, more qualified than traditional human feedback, AI itself can be used to train supervisors that perform this shaping up, provided the aim is not to eliminate 'evasiveness' as in Bai et al. (2022), but to find the right level of avoidance.

A related option that also modifies the loss function is stimulating predictability by requiring the subject to self-model, i.e. to predict their own internal states. As Premakumar et al. (2024) shows, this causes the network to "make itself more simple, regularised, + parameter-efficient, + therefore more amenable to being predictively modelled".

A second direction for increased predictability comes from recent research on mechanistic interpretability and model steering. The most popular approach has been to use sparse auto-encoders (SAEs) to learn monosemantic representations from the layers of a transformer network (Bricken et al., 2023). By integrating the trained SAEs into the model architecture, researchers can intervene on models to change their behaviour. Using predictability with respect to a single score predictor as the optimisation objective, researchers could intervene on LLM internals using SAEs to produce more predictable models.

Miscellaneous

Some of the ideas we explored in Chapter 4 would benefit from being put to practice. For example, we could 'scale the scaling laws' (section 4.2.3.2) or demonstrate an anticipative reject option for a concrete sequential decision use-case (section 4.1.3.4). Alternatively, following our definition of contamination as the instances where (good) score predictors predict bad scores while the subjects nonetheless perform well (section 3.6), we could try to quantify contamination for some specific domains and benchmarks.

This is not something we have covered before, but for learning a score predicting model we can already passively learn (i.e. learn from whatever evaluation records available) and actively learn (i.e. like adaptive testing, specifically getting the scores for instances of highest information). To complete the trifecta, it could be possible to use machine teaching (Zhu et al., 2018) techniques for evaluation to reduce the number of records needed. An analogy would contenders in the high-jump at the Olympics contenders select the bar height themselves, as opposed to it being gradually raised by a judge.

We have also differentiated prediction targets with a lot of different properties in section 4.3.3, but have given little attention to predicting losses and scores other than those derived from binary correctness. There is also some sort of nested

optimisation problem, where losses for the subject become targets for the score predictor, which might be a loss-based optimiser as well. Consider for example the subject loss being the squared error, and the score predictor loss being the squared error as well; there would be some sort of double penalisation for outliers. It is not clear how these things interact, and maybe there is something from bi-level optimisation literature (Sinha et al. (2020) or section 4.1.1.1) to borrow.

In appendix B.4, we provided multiple arguments for why score prediction might be impossible, or at least as hard as solving the original task. That reasoning can be fleshed out, and potentially connected to existential safety, considering a subject only safe to deploy if scores can be guaranteed to some very high degree of confidence to be not catastrophic. Related to both the above and the discussion on different prediction targets (section 4.3.3), there is a lot political and societal necessity for predicting risk. Not risk as in 'empirical risk measured by losses', but as in 'high-risk AI systems' as would be used in e.g. the European Commission's AI Act (European Commission, 2024). Most scores usually sum both benefit and harm, but they could be split as well. To take e.g. poem generation as a banal example, there might be several aspects of the poem you like, e.g. the metaphors and meaning, while disliking others, e.g. the rhyming and metre. If we extrapolate to more impactful tasks and a society wide distribution of AI systems, the predicted sum total score might be positive, but the risk and rewards might not be divided equally between demographics, which is something we would want to disentangle.

Lastly, as so much weight is carried by human score prediction, it would be interesting to investigate tools that not necessarily replace, but support human mental models of performance. For example, what is needed from error analysis section 4.2.5.5 to empower developers, users, policy makers to understand performance of the current generation of general-purpose AI systems?

Closing Notes

Artificial intelligence evaluation is not out of a rut yet. Good out-of-distribution performance is hard to achieve, whether predicting scores, classifying images, or anything else. Every statistic, from a simple summary statistic to an assessor model over millions of evaluation records, is tailored to a particular distribution of inputs. Naturally, this is something we have not solved. But by drawing parallels between artificial intelligence and its evaluation, we have opened the way for borrowing tools from one to the other, e.g. using meta-learning or generalist pre-training to allow for score predictors to data-efficiently adapt to new settings.

Plenty of sciences progress together with their measurement tools. As the telescope improved, so did astronomy. As the microscope improved, so did biology. We expect artificial intelligence to be no different. The evaluation of a processing is the 'learning about what has been learned'; so as the theory of learning evolved from statistics to machine learning, maybe so should the science of evaluation. Then, as models stop being sensitive to small perturbations, we can adapt the same improvement to the same affect. Maybe more than using them for anything else, we should let machines help us understand machines, so that as artificial intelligence improves, our understanding improves with it.

And keep publishing instance-level results!

Bibliography

Aenigmatis. (2019). Word shuffle: Manageable anagram puzzles. *ISBN:9781795637268* (cit. on p. 228).

Alabdulmohsin, I., Neyshabur, B., & Zhai, X. (2022). Revisiting Neural Scaling Laws in Language and Vision. Retrieved January 24, 2023, from https://openreview.net/forum?id=h3RYh6IBBS (cit. on p. 78).

Alabdulmohsin, I., Tran, V. Q., & Dehghani, M. (2024, May 22). *Fractal Patterns May Illuminate the Success of Next-Token Prediction*. arXiv: 2402.01825 [cs]. https://doi.org/10.48550/arXiv.2402.01825 (cit. on p. 89).

American Educational Research Association, American Psychological Association, & National Council on Measurement in Education. (2014). *Standards for educational and psychological testing*. (Cit. on p. 80).

Amezaga, N., & Hajek, J. (2022). Availability of Voice Deepfake Technology and its Impact for Good and Evil. *Proceedings of the 23rd Annual Conference on Information Technology Education*, 23–28. https://doi.org/10.1145/3537674.3554742 (cit. on p. 1).

Ammons, R., & Ammons, C. H. (1959). A standard anagram task. *Psychological Reports*, *5*(3), 654–656 (cit. on p. 228).

Anagrams Print. (2021). Anagrams book for adults: Funny activity book for adults. *ISBN:979-8545993970* (cit. on p. 228).

Anwar, U., Saparov, A., Rando, J., Paleka, D., Turpin, M., Hase, P., Lubana, E. S., Jenner, E., Casper, S., Sourbut, O., Edelman, B. L., Zhang, Z., Günther, M., Korinek, A., Hernandez-Orallo, J., Hammond, L., Bigelow, E., Pan, A., Langosco, L., ... Krueger, D. (2024, April 15). *Foundational Challenges in Assuring Alignment and Safety of Large Language Models*. arXiv: 2404.09932 [cs]. https://doi.org/10.48550/arXiv.2404.09932 (cit. on p. 154).

Arjovsky, M., Bottou, L., Gulrajani, I., & Lopez-Paz, D. (2020, March 27). *Invariant Risk Minimization*. Retrieved March 10, 2021, from http://arxiv.org/abs/1907.02893 (cit. on pp. 50, 163).

Arora, S., Ge, R., Halpern, Y., Mimno, D., Moitra, A., Sontag, D., Wu, Y., & Zhu, M. (2013). A Practical Algorithm for Topic Modeling with Provable Guarantees. *Proceedings of the 30th International Conference on Machine Learning*, 280–288. Retrieved September 27, 2024, from https://proceedings.mlr.press/v28/arora13.html (cit. on p. 91).

Artemenko, C., Soltanlou, M., Dresler, T., Ehlis, A.-C., & Nuerk, H.-C. (2018). The neural correlates of arithmetic difficulty depend on mathematical ability: Evidence from combined fNIRS and ERP. *Brain structure and function*, *223*(6), 2561–2574 (cit. on p. 236).

Ashcraft, M. H. (1995). Cognitive psychology and simple arithmetic: A review and summary of new directions. *Mathematical cognition*, *1*(1), 3–34 (cit. on p. 236).

Baan, J., Daheim, N., Ilia, E., Ulmer, D., Li, H.-S., Fernández, R., Plank, B., Sennrich, R., Zerva, C., & Aziz, W. (2023, July 28). *Uncertainty in Natural Language Generation: From Theory to Applications*. arXiv: 2307.15703 [cs]. Retrieved August 7, 2023, from http://arxiv.org/abs/2307.15703 (cit. on p. 16).

Baek, C., Jiang, Y., Raghunathan, A., & Kolter, J. Z. (2022). Agreement-on-the-line: Predicting the Performance of Neural Networks under Distribution Shift. *Advances in Neural Information Processing Systems*, *35*, 19274–19289. Retrieved September 11, 2024, from https://proceedings.neurips.

cc / paper _ files / paper / 2022 / hash / 7a8d388b7a17df480856dff1cc079b08 - Abstract-Conference.html (cit. on p. 62).

Bai, Y., Kadavath, S., Kundu, S., Askell, A., Kernion, J., Jones, A., Chen, A., Goldie, A., Mirhoseini, A., McKinnon, C., Chen, C., Olsson, C., Olah, C., Hernandez, D., Drain, D., Ganguli, D., Li, D., Tran-Johnson, E., Perez, E., ... Kaplan, J. (2022, December 15). *Constitutional AI: Harmlessness from AI Feedback*. arXiv: 2212.08073 [cs]. https://doi.org/10.48550/arXiv.2212.08073 (cit. on pp. 126, 128, 165, 261).

Balloccu, S., Schmidtová, P., Lango, M., & Dusek, O. (2024, March). Leak, Cheat, Repeat: Data Contamination and Evaluation Malpractices in Closed-Source LLMs. In Y. Graham & M. Purver (Eds.), *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 67–93). Association for Computational Linguistics. Retrieved September 29, 2024, from https://aclanthology.org/2024.eacl-long.5 (cit. on pp. 4, 27).

Balloccu, S., Schmidtová, P., Lango, M., & Dušek, O. (2024). Leak, cheat, repeat: Data contamination and evaluation malpractices in closed-source llms. *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics* (cit. on p. 229).

Balog, K., & Zhai, C. (2023). User Simulation for Evaluating Information Access Systems. *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, 302–305. https://doi.org/10.1145/3624918.3629549 (cit. on p. 154).

Bansal, G., Nushi, B., Kamar, E., Horvitz, E., & Weld, D. S. (2021). Is the Most Accurate AI the Best Teammate? Optimizing AI for Teamwork. *Proceedings of the AAAI Conference on Artificial Intelligence*, *35*(13), 11405–11414. https://doi.org/10.1609/aaai.v35i13.17359 (cit. on pp. 125, 128, 260).

Bansal, G., Nushi, B., Kamar, E., Lasecki, W. S., Weld, D. S., & Horvitz, E. (2019). Beyond Accuracy: The Role of Mental Models in Human-AI Team Performance. *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, *7*, 2–11. Retrieved June 16, 2022, from https://ojs.aaai.org/index.php/HCOMP/article/view/5285 (cit. on pp. 125, 128, 260).

Barnes, E. C. (2022). Prediction versus Accommodation. In E. N. Zalta & U. Nodelman (Eds.), *The Stanford Encyclopedia of Philosophy* (Winter 2022). Metaphysics Research Lab, Stanford University. Retrieved May 23, 2024, from https://plato.stanford.edu/archives/win2022/entries/prediction-accommodation/ (cit. on p. 3).

Bastani, O. (2021). Safe Reinforcement Learning with Nonlinear Dynamics via Model Predictive Shielding. *2021 American Control Conference (ACC)*, 3488–3494. https://doi.org/10.23919/ACC50511.2021.9483182 (cit. on pp. 66, 91).

Bella, A., Ferri, C., Hernández-Orallo, J., & Ramírez-Quintana, M. J. (2010). Calibration of Machine Learning Models. In *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques* (pp. 128–146). IGI Global. Retrieved August 19, 2024, from https://www.semanticscholar.org/paper/Chapter-6-Calibration-of-Machine-Learning-Models-Bella-Ferri/3284cce500a3138fac2a0b9802325589a6a75984 (cit. on p. 17).

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., & Vaughan, J. W. (2010). A theory of learning from different domains. *Machine Learning*, *79*(1-2), 151–175. https://doi.org/10.1007/s10994-009-5152-4 (cit. on p. 82).

Bengio, E., Bacon, P.-L., Pineau, J., & Precup, D. (2016, January 7). *Conditional Computation in Neural Networks for faster models*. arXiv: 1511.06297 [cs]. https://doi.org/10.48550/arXiv.1511.06297 (cit. on pp. 65, 66).

Bhatt, U., Antorán, J., Zhang, Y., Liao, Q. V., Sattigeri, P., Fogliato, R., Melançon, G., Krishnan, R., Stanley, J., Tickoo, O., Nachman, L., Chunara, R., Srikumar, M., Weller, A., & Xiang, A. (2021). Uncertainty as a Form of Transparency: Measuring, Communicating, and Using Uncertainty. *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, 401–413. https://doi.org/10.1145/3461702.3462571 (cit. on p. 91).

Bhatt, V., Tjanaka, B., Fontaine, M. C., & Nikolaidis, S. (2022, June 8). *Deep Surrogate Assisted Generation of Environments*. arXiv: 2206.04199 [cs]. https://doi.org/10.48550/arXiv.2206.04199 (cit. on pp. 59, 86, 221).

BigScience, Scao, T. L., Fan, A., Akiki, C., Pavlick, E., Ilić, S., Hesslow, D., Castagné, R., Luccioni, A. S., Yvon, F., Gallé, M., Tow, J., Rush, A. M., Biderman, S., Webson, A., Ammanamanchi, P. S., Wang, T., Sagot, B., Muen-

nighoff, N., ... Wolf, T. (2023, March 13). *BLOOM: A 176B-Parameter Open-Access Multilingual Language Model*. arXiv: 2211.05100 [cs]. https://doi.org/10.48550/arXiv.2211.05100 (cit. on pp. 128, 148, 227).

Birhane, A., Kasirzadeh, A., Leslie, D., & Wachter, S. (2023). Science in the age of large language models. *Nature Reviews Physics*, *5*(5), 277–280. https://doi.org/10.1038/s42254-023-00581-4 (cit. on pp. 125, 144).

Boiko, D. A., MacKnight, R., Kline, B., & Gomes, G. (2023a). Autonomous chemical research with large language models. *Nature*, *624*(7992), 570–578 (cit. on p. 261).

Boiko, D. A., MacKnight, R., Kline, B., & Gomes, G. (2023b). Autonomous chemical research with large language models. *Nature*, *624*(7992), 570–578. https://doi.org/10.1038/s41586-023-06792-0 (cit. on p. 144).

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., Brynjolfsson, E., Buch, S., Card, D., Castellon, R., Chatterji, N., Chen, A., Creel, K., Davis, J. Q., Demszky, D., ... Liang, P. (2021, August 18). *On the Opportunities and Risks of Foundation Models*. Retrieved August 23, 2021, from http://arxiv.org/abs/2108.07258 (cit. on p. 14).

Bommasani, R., Soylu, D., Liao, T. I., Creel, K. A., & Liang, P. (2023, March 28). *Ecosystem Graphs: The Social Footprint of Foundation Models*. arXiv: 2303.15772 [cs]. https://doi.org/10.48550/arXiv.2303.15772 (cit. on pp. 127, 143).

Bowden, E. M. (1997). The effect of reportable and unreportable hints on anagram solution and the aha! experience. *Consciousness and cognition*, *6*(4), 545–573 (cit. on p. 228).

Breiman, L. (2001). Statistical Modeling: The Two Cultures. *Statistical Science*, *16*(3), 199–231. https://doi.org/10.1214/ss/1009213726 (cit. on pp. 4, 67, 161).

Bricken, T., Templeton, A., Batson, J., Chen, B., Jermyn, A., & Olah, C. (2023, April 10). *Towards Monosemanticity: Decomposing Language Models With Dictionary Learning*. Anthropic. https://transformer-circuits.pub/2023/monosemantic-features (cit. on p. 165).

Brier, G. W. (1950). Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, *78*(1), 1–3. https://doi.org/10.1175/1520-0493(1950)078<0001:VOFEIT>2.0.CO;2 (cit. on pp. 17, 111).

Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Chen, X., Choromanski, K., Ding, T., Driess, D., Dubey, A., Finn, C., Florence, P., Fu, C., Arenas, M. G., Gopalakrishnan, K., Han, K., Hausman, K., Herzog, A., Hsu, J., Ichter, B., … Zitkovich, B. (2023, July 26). *RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control*. Google Deepmind. https://deepmind.google/discover/blog/rt-2-new-model-translates-vision-and-language-into-action/ (cit. on p. 67).

Bronstein, E., Srinivasan, S., Paul, S., Sinha, A., O'Kelly, M., Nikdel, P., & Whiteson, S. (2022). Embedding Synthetic Off-Policy Experience for Autonomous Driving via Zero-Shot Curricula. Retrieved December 13, 2022, from https://openreview.net/forum?id=cF1dxVGxic- (cit. on pp. 59, 83).

Brooks, R. (2021, November 3). *Artificial Intimacy: Virtual Friends, Digital Lovers, and Algorithmic Matchmakers*. Columbia University Press. https://doi.org/10.7312/broo20094 (cit. on p. 1).

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., … Amodei, D. (2020). Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems*, *33*, 1877–1901. Retrieved September 25, 2022, from https://papers.nips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html (cit. on pp. 125, 148, 227).

Browne, M. W. (2000). Cross-Validation Methods. *Journal of Mathematical Psychology*, *44*(1), 108–132. https://doi.org/10.1006/jmps.1999.1279 (cit. on pp. 19, 27, 73).

Browning, J., & LeCun, Y. (2023). Language, Common Sense, and the Winograd Schema Challenge. *Artificial Intelligence*, 104031. https://doi.org/10.1016/j.artint.2023.104031 (cit. on p. 154).

Burden, J., Voudouris, K., Burnell, R., Rutar, D., Cheke, L., & Hernández-Orallo, J. (2023, September 21). *Inferring Capabilities from Task Performance with*

Bayesian Triangulation. arXiv: 2309.11975 [cs]. Retrieved September 26, 2023, from http://arxiv.org/abs/2309.11975 (cit. on pp. 88, 92).

Burnell, R., Burden, J., Rutar, D., Voudouris, K., Cheke, L., & Hernandez-Orallo, J. (2022). Not a Number: Identifying Instance Features for Capability-Oriented Evaluation, 9 (cit. on pp. 84, 223).

Burnell, R., Hao, H., Conway, A. R. A., & Orallo, J. H. (2023, June 14). *Revealing the structure of language model capabilities*. arXiv: 2306.10062 [cs]. Retrieved September 11, 2023, from http://arxiv.org/abs/2306.10062 (cit. on p. 74).

Burnell, R., Schellaert, W., Burden, J., Ullman, T. D., Martinez-Plumed, F., Tenenbaum, J. B., Rutar, D., Cheke, L. G., Sohl-Dickstein, J., Mitchell, M., Kiela, D., Shanahan, M., Voorhees, E. M., Cohn, A. G., Leibo, J. Z., & Hernandez-Orallo, J. (2023). Rethink reporting of evaluation results in AI. *Science*, *380*(6641), 136–138. https://doi.org/10.1126/science.adf6369 (cit. on pp. 7, 20, 24, 112, 150, 153, 218).

Caballero, E., Gupta, K., Rish, I., & Krueger, D. (2022). Broken Neural Scaling Laws. Retrieved October 10, 2023, from https://openreview.net/forum?id=sckjveqlCZ (cit. on pp. 58, 78).

Campero, A., Raileanu, R., Küttler, H., Tenenbaum, J. B., Rocktäschel, T., & Grefenstette, E. (2021, February 23). *Learning with AMIGo: Adversarially Motivated Intrinsic Goals*. arXiv: 2006.12122 [cs, stat]. https://doi.org/10.48550/arXiv.2006.12122 (cit. on p. 59).

Carlini, N. (2023). *A GPT-4 Capability Forecasting Challenge*. Retrieved August 18, 2023, from https://nicholas.carlini.com/writing/llm-forecast/ (cit. on pp. 4, 26, 89).

Carlini, N., & Wagner, D. (2017). Towards Evaluating the Robustness of Neural Networks. *2017 IEEE Symposium on Security and Privacy (SP)*, 39–57. https://doi.org/10.1109/SP.2017.49 (cit. on pp. 20, 91).

Celis, L. E., Huang, L., Keswani, V., & Vishnoi, N. K. (2019). Classification with Fairness Constraints: A Meta-Algorithm with Provable Guarantees. *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 319–328. https://doi.org/10.1145/3287560.3287586 (cit. on p. 91).

Chen, H., Lundberg, S. M., & Lee, S.-I. (2022). Explaining a series of models by propagating Shapley values. *Nature Communications*, *13*(1), 4512. https://doi.org/10.1038/s41467-022-31384-3 (cit. on p. 68).

Chen, L., Zaharia, M., & Zou, J. (2022). Efficient Online ML API Selection for Multi-Label Classification Tasks. *Proceedings of the 39th International Conference on Machine Learning*, 3716–3746. Retrieved September 12, 2024, from https://proceedings.mlr.press/v162/chen22ad.html (cit. on p. 63).

Chen, L., Zaharia, M., & Zou, J. (2024). How Is ChatGPT's Behavior Changing Over Time? *Harvard Data Science Review*, *6*(2). https://doi.org/10.1162/99608f92.5317da47 (cit. on pp. 127, 260).

Chen, Y., Yuan, L., Cui, G., Liu, Z., & Ji, H. (2022, November 28). *A Close Look into the Calibration of Pre-trained Language Models*. arXiv: 2211.00151 [cs]. https://doi.org/10.48550/arXiv.2211.00151 (cit. on p. 17).

Cheng, A.-C., Lin, C. H., Juan, D.-C., Wei, W., & Sun, M. (2020). InstaNAS: Instance-Aware Neural Architecture Search. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(04), 3577–3584. https://doi.org/10.1609/aaai.v34i04.5764 (cit. on pp. 57, 90).

Chiang, W.-L., Zheng, L., Sheng, Y., Angelopoulos, A. N., Li, T., Li, D., Zhang, H., Zhu, B., Jordan, M., Gonzalez, J. E., & Stoica, I. (2024, March 6). *Chatbot Arena: An Open Platform for Evaluating LLMs by Human Preference*. arXiv: 2403.04132 [cs]. https://doi.org/10.48550/arXiv.2403.04132 (cit. on pp. 26, 48, 61, 90).

Choi, B., Ward, A., Li, Y., Arguello, J., & Capra, R. (2019). The effects of task complexity on the use of different types of information in a search assistance tool. *ACM Transactions on Information Systems (TOIS)*, *38*(1), 1–28 (cit. on p. 239).

Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S., et al. (2024). Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, *25*(70), 1–53 (cit. on p. 227).

Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S., Webson, A., Gu, S. S., Dai, Z., Suzgun,

M., Chen, X., Chowdhery, A., Castro-Ros, A., Pellat, M., Robinson, K., … Wei, J. (2022, December 6). *Scaling Instruction-Finetuned Language Models*. arXiv: 2210.11416 [cs]. https://doi.org/10.48550/arXiv.2210.11416 (cit. on pp. 14, 125).

Clark, A., & Chalmers, D. (1998). The Extended Mind. *Analysis*, *58*(1), 7–19. Retrieved June 1, 2023, from https://www.jstor.org/stable/3328150 (cit. on p. 42).

Clark, K., Luong, M.-T., Le, Q. V., & Manning, C. D. (2019). ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. Retrieved August 19, 2024, from https://openreview.net/forum?id=r1xMH1BtvB (cit. on p. 109).

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., & Schulman, J. (2021, November 17). *Training Verifiers to Solve Math Word Problems*. arXiv: 2110.14168 [cs]. https://doi.org/10.48550/arXiv.2110.14168 (cit. on p. 91).

Cohen, J. L. (1968). The effect of letter frequency on anagram solution times. *Psychonomic Science*, *11*(2), 79–80 (cit. on p. 237).

Collins, K. M., Jiang, A. Q., Frieder, S., Wong, L., Zilka, M., Bhatt, U., Lukasiewicz, T., Wu, Y., Tenenbaum, J. B., Hart, W., et al. (2024). Evaluating language models for mathematics through interactions. *Proceedings of the National Academy of Sciences*, *121*(24), e2318124121 (cit. on p. 260).

Collins, K. M., Jiang, A. Q., Frieder, S., Wong, L., Zilka, M., Bhatt, U., Lukasiewicz, T., Wu, Y., Tenenbaum, J. B., Hart, W., Gowers, T., Li, W., Weller, A., & Jamnik, M. (2023, June 2). *Evaluating Language Models for Mathematics through Interactions*. arXiv: 2306.01694 [cs]. https://doi.org/10.48550/arXiv.2306.01694 (cit. on p. 154).

Condessa, F., Bioucas-Dias, J., & Kovačević, J. (2017). Performance measures for classification systems with rejection. *Pattern Recognition*, *63*, 437–450. https://doi.org/10.1016/j.patcog.2016.10.011 (cit. on pp. 40, 41).

Corbiere, C., Thome, N., Saporta, A., Vu, T.-H., Cord, M., & Perez, P. (2021). Confidence Estimation via Auxiliary Models. *IEEE Transactions on Pattern*

Analysis and Machine Intelligence, 1–1. https://doi.org/10.1109/TPAMI.2021.3085983 (cit. on pp. 90, 97, 102).

Corbière, C., THOME, N., Bar-Hen, A., Cord, M., & Pérez, P. (2019). Addressing Failure Prediction by Learning Model Confidence. *Advances in Neural Information Processing Systems*, *32*. Retrieved September 15, 2021, from https://papers.nips.cc/paper/2019/hash/757f843a169cc678064d9530d12a1881-Abstract.html (cit. on pp. 90, 102).

Costa, D. C. da, Prudêncio, R., & Mota, A. (2023). Assessor Models with Reject Option for soccer result prediction. *Anais do Encontro Nacional de Inteligência Artificial e Computacional (ENIAC)*, 683–696. https://doi.org/10.5753/eniac.2023.234321 (cit. on pp. 86, 105).

Covington, M. V. (1984). The Self-Worth Theory of Achievement Motivation: Findings and Implications. *The Elementary School Journal*, *85*(1), 5–20. Retrieved August 19, 2024, from https://www.jstor.org/stable/1001615 (cit. on pp. 128, 261).

Dai, D., Deng, C., Zhao, C., Xu, R. X., Gao, H., Chen, D., Li, J., Zeng, W., Yu, X., Wu, Y., Xie, Z., Li, Y. K., Huang, P., Luo, F., Ruan, C., Sui, Z., & Liang, W. (2024, January 11). *DeepSeekMoE: Towards Ultimate Expert Specialization in Mixture-of-Experts Language Models*. arXiv: 2401.06066 [cs]. https://doi.org/10.48550/arXiv.2401.06066 (cit. on p. 163).

De Cao, N., Aziz, W., & Titov, I. (2021). Editing Factual Knowledge in Language Models. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 6491–6506. https://doi.org/10.18653/v1/2021.emnlp-main.522 (cit. on p. 123).

de Ayala, R. J. (2009). *The theory and practice of item response theory*. Guilford Press. (Cit. on pp. 21, 76).

Deng, B., Yan, J., & Lin, D. (2017, December 9). *Peephole: Predicting Network Performance Before Training*. arXiv: 1712.03351 [cs, stat]. https://doi.org/10.48550/arXiv.1712.03351 (cit. on pp. 57, 81).

Deng, C., Zhao, Y., Tang, X., Gerstein, M., & Cohan, A. (2024, June). Investigating data contamination in modern benchmarks for large language models. 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 (volume 1: Long papers) (pp. 8698–8711). Association for Computational Linguistics. https://aclanthology.org/2024.naacl-long.482 (cit. on p. 261).

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–255. https://doi.org/10.1109/CVPR.2009.5206848 (cit. on p. 63).

Deng, W., Gould, S., & Zheng, L. (2021). What Does Rotation Prediction Tell Us about Classifier Accuracy under Varying Testing Environments? *Proceedings of the 38th International Conference on Machine Learning*, 2579–2589. Retrieved December 9, 2021, from https://proceedings.mlr.press/v139/deng21a.html (cit. on p. 63).

Dennis, M., Jaques, N., Vinitsky, E., Bayen, A., Russell, S., Critch, A., & Levine, S. (2020). Emergent complexity and zero-shot transfer via unsupervised environment design. *Proceedings of the 34th International Conference on Neural Information Processing Systems*, 13049–13061 (cit. on p. 59).

Deschuyteneer, M., De Rammelaere, S., & Fias, W. (2005). The addition of two-digit numbers: Exploring carry versus no-carry problems. *Psychology Science*, *47*(1), 74–83 (cit. on p. 236).

Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2023). QLoRA: Efficient Finetuning of Quantized LLMs. Retrieved August 19, 2024, from https://openreview.net/forum?id=OUIFPHEgJU (cit. on p. 148).

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186. https://doi.org/10.18653/v1/N19-1423 (cit. on pp. 14, 64, 110, 125).

Díaz-Rodríguez, N., Del Ser, J., Coeckelbergh, M., López de Prado, M., Herrera-Viedma, E., & Herrera, F. (2023). Connecting the dots in trustworthy Artificial Intelligence: From AI principles, ethics, and key requirements to re-

sponsible AI systems and regulation. *Information Fusion*, *99*, 101896. https://doi.org/10.1016/j.inffus.2023.101896 (cit. on p. 154).

Dimitriadis, T., Gneiting, T., & Jordan, A. I. (2021). Stable reliability diagrams for probabilistic classifiers. *Proceedings of the National Academy of Sciences*, *118*(8), e2016191118. https://doi.org/10.1073/pnas.2016191118 (cit. on pp. 17, 111).

Ding, D., Mallick, A., Wang, C., Sim, R., Mukherjee, S., Rühle, V., Lakshmanan, L. V. S., & Awadallah, A. H. (2023). Hybrid LLM: Cost-Efficient and Quality-Aware Query Routing. Retrieved July 2, 2024, from https://openreview.net/forum?id=02f3mUtqnM (cit. on pp. 64, 102, 103).

Domhan, T., Springenberg, J. T., & Hutter, F. (2015). Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves. *Proceedings of the 24th International Conference on Artificial Intelligence*, 3460–3468 (cit. on p. 78).

Dominguez-Olmedo, R., Dorner, F. E., & Hardt, M. (2024, July 10). *Training on the Test Task Confounds Evaluation and Emergence*. arXiv: 2407.07890 [cs]. Retrieved September 8, 2024, from http://arxiv.org/abs/2407.07890 (cit. on p. 27).

Dominowski, R. L., & Ekstrand, B. R. (1967). Direct and associative priming in anagram solving. *Journal of Experimental Psychology*, *74*(1), 84 (cit. on p. 228).

Douglas, H. E. (2009). Reintroducing Prediction to Explanation. *Philosophy of Science*, *76*(4), 444–463. https://doi.org/10.1086/648111 (cit. on p. 3).

Drapal, P., Silva-Filho, T., & Prudêncio, R. B. C. (2024). Meta-Learning and Novelty Detection for Machine Learning with Reject Option. *2024 International Joint Conference on Neural Networks (IJCNN)*, 1–8. https://doi.org/10.1109/IJCNN60899.2024.10650424 (cit. on pp. 66, 85, 86).

Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., Goyal, A., Hartshorn, A., Yang, A., Mitra, A., Sravankumar, A., Korenev, A., Hinsvark, A., Rao, A., Zhang, A., … Zhao, Z. (2024, August 15). *The Llama 3 Herd of Models*. arXiv: 2407.21783 [cs]. https://doi.org/10.48550/arXiv.2407.21783 (cit. on p. 163).

Duhem, P., Wiener, P. P., & Vuillemin, J. (1982). *The Aim and Structure of Physical Theory* (Vol. 126). Princeton University Press. https://doi.org/10.2307/j.ctv1nj34vm (cit. on pp. 2, 53).

Dweck, C. S., & Bempechat, J. (2017). Children's theories of intelligence: Consequences for learning. In *Learning and motivation in the classroom* (pp. 239–256). Routledge. (Cit. on p. 261).

Ekstrand, M. D., Riedl, J. T., & Konstan, J. A. (2011). Collaborative Filtering Recommender Systems. *Foundations and Trends® in Human–Computer Interaction*, *4*(2), 81–173. https://doi.org/10.1561/1100000009 (cit. on p. 96).

Elliot, S. (2023, July 15). *Putting AI to the test: How does the performance of GPT and 15-year-old students in PISA compare?* (OECD Education Spotlights No. 6). OECD. https://doi.org/10.1787/2c297e0b-en (cit. on p. 218).

Eloundou, T., Manning, S., Mishkin, P., & Rock, D. (2023, August 21). *GPTs are GPTs: An Early Look at the Labor Market Impact Potential of Large Language Models*. arXiv: 2303.10130 [cs, econ, q-fin]. https://doi.org/10.48550/arXiv.2303.10130 (cit. on pp. 14, 125).

Elsahar, H., & Gallé, M. (2019). To Annotate or Not? Predicting Performance Drop under Domain Shift. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2163–2173. https://doi.org/10.18653/v1/D19-1222 (cit. on p. 63).

Elsken, T., Metzen, J. H., & Hutter, F. (2019). Neural architecture search: A survey. *J. Mach. Learn. Res.*, *20*(1), 1997–2017 (cit. on p. 55).

Erdogan, L. E., Lee, N., Jha, S., Kim, S., Tabrizi, R., Moon, S., Hooper, C., Anumanchipalli, G., Keutzer, K., & Gholami, A. (2024, September 1). *TinyAgent: Function Calling at the Edge*. arXiv: 2409.00608 [cs]. Retrieved September 21, 2024, from http://arxiv.org/abs/2409.00608 (cit. on p. 65).

Ethayarajh, K., Choi, Y., & Swayamdipta, S. (2022). Understanding Dataset Difficulty with $\mathcal{V}$-Usable Information. *Proceedings of the 39th International Conference on Machine Learning*, 5988–6008. Retrieved November 29, 2022, from https://proceedings.mlr.press/v162/ethayarajh22a.html (cit. on pp. 39, 74).

AI Act (2024). Retrieved May 19, 2024, from https://www.europarl.europa.eu/doceo/document/TA-9-2024-0138-FNL-COR01_EN.pdf (cit. on pp. 7, 166).

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, *27*(8), 861–874. https://doi.org/10.1016/j.patrec.2005.10.010 (cit. on p. 19).

Feather, N. T. (Ed.). (2021, December 30). *Expectations and Actions: Expectancy-Value Models in Psychology*. Routledge. https://doi.org/10.4324/9781003150879 (cit. on pp. 125, 128).

Fedus, W., Dean, J., & Zoph, B. (2022, September 4). *A Review of Sparse Expert Models in Deep Learning*. arXiv: 2209.01667 [cs]. https://doi.org/10.48550/arXiv.2209.01667 (cit. on pp. 65, 123).

Filho, T. S., Song, H., Perello-Nieto, M., Santos-Rodriguez, R., Kull, M., & Flach, P. (2021, December 19). *Classifier Calibration: A Survey*. arXiv: 2112.10327. https://doi.org/10.48550/arXiv.2112.10327 (cit. on pp. 16, 17).

Fischer, L., & Wollstadt, P. (2023). Precision and recall reject curves for classification. *arXiv preprint arXiv:2308.08381* (cit. on p. 261).

Florensa, C., Held, D., Geng, X., & Abbeel, P. (2018). Automatic Goal Generation for Reinforcement Learning Agents. *Proceedings of the 35th International Conference on Machine Learning*, 1515–1528. Retrieved September 13, 2024, from https://proceedings.mlr.press/v80/florensa18a.html (cit. on p. 59).

Franceschi, L., Frasconi, P., Salzo, S., Grazzi, R., & Pontil, M. (2018, July 3). *Bilevel Programming for Hyperparameter Optimization and Meta-Learning*. arXiv: 1806.04910 [cs, stat]. Retrieved September 10, 2024, from http://arxiv.org/abs/1806.04910 (cit. on p. 55).

Franz, A., & Brants, T. (2006). *All Our N-gram are Belong to You*. Retrieved August 19, 2024, from http://research.google/blog/all-our-n-gram-are-belong-to-you/ (cit. on p. 146).

Frazier, P. I. (2018, July 8). *A Tutorial on Bayesian Optimization*. arXiv: 1807.02811 [cs, math, stat]. https://doi.org/10.48550/arXiv.1807.02811 (cit. on p. 55).

Frenay, B., & Verleysen, M. (2014). Classification in the Presence of Label Noise: A Survey. *IEEE Transactions on Neural Networks and Learning Systems*, *25*(5), 845–869. https://doi.org/10.1109/TNNLS.2013.2292894 (cit. on p. 163).

Frey, L. J., & Fisher, D. H. (1999). Modeling decision tree performance with the power law. *Seventh International Workshop on Artificial Intelligence and Statistics*. Retrieved September 27, 2024, from https://proceedings.mlr.press/r2/frey99a.html (cit. on pp. 57, 58, 78).

Garg, S., Balakrishnan, S., Lipton, Z. C., Neyshabur, B., & Sedghi, H. (2021). Leveraging unlabeled data to predict out-of-distribution performance. Retrieved April 5, 2022, from https://openreview.net/forum?id=o_HsiMPYh_x (cit. on p. 226).

Gehrmann, S., Clark, E., & Sellam, T. (2023). Repairing the Cracked Foundation: A Survey of Obstacles in Evaluation Practices for Generated Text. *J. Artif. Int. Res.*, *77*. https://doi.org/10.1613/jair.1.13715 (cit. on p. 9).

Gencoglu, O., van Gils, M., Guldogan, E., Morikawa, C., Süzen, M., Gruber, M., Leinonen, J., & Huttunen, H. (2019, April 16). *HARK Side of Deep Learning – From Grad Student Descent to Automated Machine Learning*. arXiv: 1904.07633 [cs]. https://doi.org/10.48550/arXiv.1904.07633 (cit. on p. 23).

Ghorbani, A., & Zou, J. (2019). Data Shapley: Equitable Valuation of Data for Machine Learning. *Proceedings of the 36th International Conference on Machine Learning*, 2242–2251. Retrieved September 14, 2024, from https://proceedings.mlr.press/v97/ghorbani19c.html (cit. on p. 68).

Gilhooly, K., & Johnson, C. (1978). Effects of solution word attributes on anagram difficulty: A regression analysis. *Quarterly Journal of Experimental Psychology*, *30*(1), 57–70 (cit. on p. 228).

Gneiting, T., & Raftery, A. E. (2007). Strictly Proper Scoring Rules, Prediction, and Estimation. *Journal of the American Statistical Association*, *102*(477), 359–378. https://doi.org/10.1198/016214506000001437 (cit. on p. 16).

Göbel, S. M., Moeller, K., Pixner, S., Kaufmann, L., & Nuerk, H.-C. (2014). Language affects symbolic arithmetic in children: The case of number word inversion. *Journal of experimental child psychology*, *119*, 17–25 (cit. on p. 236).

Golchin, S., & Surdeanu, M. (2024). Time travel in LLMs: Tracing data contamination in large language models. *The Twelfth International Conference on Learning Representations* (cit. on p. 229).

Goode, M. K., Geraci, L., & Roediger, H. L. (2008). Superiority of variable to repeated practice in transfer on anagram solution. *Psychonomic Bulletin & Review*, *15*, 662–666 (cit. on p. 228).

Grąbczewski, K. (2014). *Meta-Learning in Decision Tree Induction* (Vol. 498). Springer International Publishing. https://doi.org/10.1007/978-3-319-00960-5 (cit. on pp. 57, 85, 163).

Grömping, U. (2007). Estimators of Relative Importance in Linear Regression Based on Variance Decomposition. *The American Statistician*, *61*(2), 139–147. Retrieved September 14, 2024, from https://www.jstor.org/stable/27643865 (cit. on p. 68).

Grömping, U. (2009). Variable Importance Assessment in Regression: Linear Regression versus Random Forest. *The American Statistician*, *63*(4), 308–319. Retrieved September 14, 2024, from https://www.jstor.org/stable/25652309 (cit. on p. 68).

Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017, August 3). *On Calibration of Modern Neural Networks*. Retrieved May 27, 2021, from http://arxiv.org/abs/1706.04599 (cit. on p. 17).

Gur, I., Jaques, N., Miao, Y., Choi, J., Tiwari, M., Lee, H., & Faust, A. (2021). Environment Generation for Zero-Shot Compositional Reinforcement Learning. Retrieved December 7, 2021, from https://openreview.net/forum?id=CeByDMy0YTL (cit. on p. 59).

Ha, H., Gupta, S., Rana, S., & Venkatesh, S. (2021, April 11). *ALT-MAS: A Data-Efficient Framework for Active Testing of Machine Learning Algorithms*. Retrieved July 21, 2021, from http://arxiv.org/abs/2104.04999 (cit. on p. 60).

Hafner, D., Lillicrap, T. P., Norouzi, M., & Ba, J. (2020). Mastering Atari with Discrete World Models. Retrieved January 17, 2022, from https://openreview.net/forum?id=0oabwyZbOu (cit. on p. 78).

Han, Y., Huang, G., Song, S., Yang, L., Wang, H., & Wang, Y. (2022). Dynamic Neural Networks: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(11), 7436–7456. https://doi.org/10.1109/TPAMI.2021.3117837 (cit. on pp. 65, 66).

Hansen, L. K., Liisberg, C., & Salamon, P. (1997). The Error-Reject Tradeoff. *Open Systems & Information Dynamics*, *4*(2), 159–184. https://doi.org/10.1023/A:1009643503022 (cit. on p. 41).

Hanson, N. R. (1959). On the Symmetry Between Explanation and Prediction. *The Philosophical Review*, *68*(3), 349–358. https://doi.org/10.2307/2182567 (cit. on p. 3).

Hardalov, M., Koychev, I., & Nakov, P. (2018). Towards Automated Customer Support. In G. Agre, J. van Genabith, & T. Declerck (Eds.), *Artificial Intelligence: Methodology, Systems, and Applications* (pp. 48–59). Springer International Publishing. https://doi.org/10.1007/978-3-319-99344-7_5 (cit. on p. 1).

Harter, S. (1978). Pleasure derived from challenge and the effects of receiving grades on children's difficulty level choices. *Child Development*, 788–799 (cit. on p. 237).

He, P., Gao, J., & Chen, W. (2021, December 8). *DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing*. arXiv: 2111.09543 [cs]. https://doi.org/10.48550/arXiv.2111.09543 (cit. on pp. 105, 109, 115).

He, P., Liu, X., Gao, J., & Chen, W. (2021). DeBERTa: Decoding-Enhanced BERT With Disentangled Attention. Retrieved December 9, 2022, from https://openreview.net/forum?id=XPZIaotutsD (cit. on p. 109).

He, X., Zhao, K., & Chu, X. (2021). AutoML: A survey of the state-of-the-art. *Knowledge-Based Systems*, *212*, 106622. https://doi.org/10.1016/j.knosys.2020.106622 (cit. on p. 55).

Hempel, C. G. (1942). The Function of General Laws in History. *The Journal of Philosophy*, *39*(2), 35–48. https://doi.org/10.2307/2017635 (cit. on p. 3).

Hendrickx, K., Perini, L., Van der Plas, D., Meert, W., & Davis, J. (2024). Machine learning with a reject option: A survey. *Machine Learning*, *113*(5), 3073–3110. https://doi.org/10.1007/s10994-024-06534-x (cit. on pp. 22, 40, 41, 66, 82, 98, 261).

Hernández-Orallo, J. (2017, January 11). *The Measure of All Minds: Evaluating Natural and Artificial Intelligence* (1 edition). Cambridge University Press. (Cit. on p. 80).

Hernández-Orallo, J., Loe, B. S., Cheke, L., Martínez-Plumed, F., & Ó hÉigeartaigh, S. (2021). General intelligence disentangled via a generality metric for natural and artificial intelligence. *Scientific Reports*, *11*(1), 22822. https://doi.org/10.1038/s41598-021-01997-7 (cit. on pp. 83, 150).

Hernández-Orallo, J., Schellaert, W., & Martínez-Plumed, F. (2022). Training on the Test Set: Mapping the System-Problem Space in AI. *Proceedings of the AAAI Conference on Artificial Intelligence*, *36*(11), 12256–12261. https://doi.org/10.1609/aaai.v36i11.21487 (cit. on pp. 7, 33, 159, 217).

Hitchcock, C., & Sober, E. (2004). Prediction Versus Accommodation and the Risk of Overfitting. *The British Journal for the Philosophy of Science*, *55*(1), 1–34. https://doi.org/10.1093/bjps/55.1.1 (cit. on p. 5).

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., Casas, D. de las, Hendricks, L. A., Welbl, J., Clark, A., Hennigan, T., Noland, E., Millican, K., Driessche, G. van den, Damoc, B., Guy, A., Osindero, S., Simonyan, K., Elsen, E., … Sifre, L. (2022). An empirical analysis of compute-optimal large language model training. Retrieved October 10, 2023, from https://openreview.net/forum?id=iBBcRUlOAPR (cit. on pp. 69, 78, 136).

Hofstadter, A. (1951). Explanation and Necessity. *Philosophy and Phenomenological Research*, *11*(3), 339–347. https://doi.org/10.2307/2103538 (cit. on p. 3).

Hospedales, T., Antoniou, A., Micaelli, P., & Storkey, A. (2020, November 7). *Meta-Learning in Neural Networks: A Survey*. Retrieved March 1, 2022, from http://arxiv.org/abs/2004.05439 (cit. on pp. 50, 56, 85, 163).

Hu, Q. J., Bieker, J., Li, X., Jiang, N., Keigwin, B., Ranganath, G., Keutzer, K., & Upadhyay, S. K. (2024, March 28). *RouterBench: A Benchmark for Multi-LLM Routing System* (2). arXiv: 2403.12031 [cs]. https://doi.org/10.48550/arXiv.2403.12031 (cit. on pp. 64, 86, 105, 110, 157, 163).

Huang, P.-S., He, X., Gao, J., Deng, L., Acero, A., & Heck, L. (2013). Learning deep structured semantic models for web search using clickthrough data. *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*, 2333–2338. https://doi.org/10.1145/2505515.2505665 (cit. on pp. 86, 96, 163).

Huang, X., Kwiatkowska, M., Wang, S., & Wu, M. (2017). Safety Verification of Deep Neural Networks. In R. Majumdar & V. Kunčak (Eds.), *Computer Aided Verification* (pp. 3–29). Springer International Publishing. https://doi.org/10.1007/978-3-319-63387-9_1 (cit. on p. 91).

Hüllermeier, E., & Waegeman, W. (2021). Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, *110*(3), 457–506. https://doi.org/10.1007/s10994-021-05946-3 (cit. on pp. 17, 18).

Hutter, M. (2021, February 8). *Learning Curve Theory*. arXiv: 2102.04074 [cs, stat]. https://doi.org/10.48550/arXiv.2102.04074 (cit. on pp. 77, 78).

Huyen, C. (2022). *Designing machine learning systems: An iterative process for production-ready applications* (First edition). O'Reilly Media, Inc. (Cit. on p. 2).

Ilyas, A., Park, S. M., Engstrom, L., Leclerc, G., & Madry, A. (2022, February 1). *Datamodels: Predicting Predictions from Training Data*. Retrieved April 12, 2022, from http://arxiv.org/abs/2202.00622 (cit. on pp. 32, 61, 62, 102, 164, 223).

Istrate, R., Scheidegger, F., Mariani, G., Nikolopoulos, D., Bekas, C., & Malossi, A. C. I. (2019). TAPAS: Train-Less Accuracy Predictor for Architecture Search. *Proceedings of the AAAI Conference on Artificial Intelligence*,

33(01), 3927–3934. https://doi.org/10.1609/aaai.v33i01.33013927 (cit. on pp. 57, 81, 85).

Jacobs, R., Jordan, M., Nowlan, S., & Hinton, G. (1991). Adaptive Mixture of Local Expert. *Neural Computation*, *3*, 78–88. https://doi.org/10.1162/neco.1991.3.1.79 (cit. on p. 65).

Jaimovitch-Lopez, G., Ferri, C., Hernandez-Orallo, J., Martınez-Plumed, F., & Ramırez-Quintana, M. J. (2021). Can language models automate data wrangling?, 13 (cit. on pp. 144, 146).

Jain, A., Swaminathan, G., Favaro, P., Yang, H., Ravichandran, A., Harutyunyan, H., Achille, A., Dabeer, O., Schiele, B., Swaminathan, A., & Soatto, S. (2023). A Meta-Learning Approach to Predicting Performance and Data Requirements, 3623–3632. Retrieved October 10, 2023, from https://openaccess.thecvf.com/content/CVPR2023/html/Jain_A_Meta-Learning_Approach_to_Predicting_Performance_and_Data_Requirements_CVPR_2023_paper.html (cit. on pp. 58, 85).

Jang, J., Kim, S., Ye, S., Kim, D., Logeswaran, L., Lee, M., Lee, K., & Seo, M. (2023). Exploring the benefits of training expert language models over instruction tuning. *Proceedings of the 40th International Conference on Machine Learning*, *202*, 14702–14729 (cit. on pp. 63, 85).

Jia, R., & Liang, P. (2017). Adversarial Examples for Evaluating Reading Comprehension Systems. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2021–2031. https://doi.org/10.18653/v1/D17-1215 (cit. on p. 20).

Jia, R., Dao, D., Wang, B., Hubis, F. A., Hynes, N., Gürel, N. M., Li, B., Zhang, C., Song, D., & Spanos, C. J. (2019). Towards Efficient Data Valuation Based on the Shapley Value. *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, 1167–1176. Retrieved September 14, 2024, from https://proceedings.mlr.press/v89/jia19a.html (cit. on p. 68).

Jiang, M., Liu, K., Zhong, M., Schaeffer, R., Ouyang, S., Han, J., & Koyejo, S. (2024). Does data contamination make a difference? insights from intentionally contamination pre-training data for language models. *ICLR 2024 Work-*

shop on Mathematical and Empirical Understanding of Foundation Models (cit. on p. 229).

Jiang, M., Liu, K. Z., Zhong, M., Schaeffer, R., Ouyang, S., Han, J., & Koyejo, S. (2024). Investigating data contamination for pre-training language models. *arXiv preprint arXiv:2401.06059* (cit. on p. 229).

Jiang, Y., Nagarajan, V., Baek, C., & Kolter, J. Z. (2022). Assessing Generalization of SGD via Disagreement. Retrieved February 1, 2023, from https://openreview.net/forum?id=WvOGCEAQhxl (cit. on p. 62).

Jin, Y. (2011). Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation*, *1*(2), 61–70. https://doi.org/10.1016/j.swevo.2011.05.001 (cit. on pp. 56, 85).

Johnson, D. M. (1966). Solution of anagrams. *Psychological Bulletin*, *66*(5), 371–384. https://doi.org/10.1037/h0023886 (cit. on p. 144).

Johnson, D., Goodman, R., Patrinely, J., Stone, C., Zimmerman, E., Donald, R., Chang, S., Berkowitz, S., Finn, A., Jahangir, E., et al. (2023). Assessing the accuracy and reliability of ai-generated medical responses: An evaluation of the chat-gpt model. *Research square* (cit. on p. 259).

Joint Committee for Guides in Metrology. (2008). JCGM 100:2008. Evaluation of measurement data – Guide to the expression of uncertainty in measurement. Retrieved January 3, 2025, from https://www.bipm.org/documents/20126/2071204/JCGM_100_2008_E.pdf/cb0ef43f-baa5-11cf-3f85-4dcd86f77bd6 (cit. on p. 103).

Kadavath, S., Conerly, T., Askell, A., Henighan, T., Drain, D., Perez, E., Schiefer, N., Dodds, Z. H., DasSarma, N., Tran-Johnson, E., Johnston, S., El-Showk, S., Jones, A., Elhage, N., Hume, T., Chen, A., Bai, Y., Bowman, S., Fort, S., ... Kaplan, J. (2022, July 11). *Language Models (Mostly) Know What They Know*. arXiv: 2207.05221. https://doi.org/10.48550/arXiv.2207.05221 (cit. on pp. 120, 127, 259).

Kag, A., Fedorov, I., Gangrade, A., Whatmough, P., & Saligrama, V. (2022). Efficient Edge Inference by Selective Query. Retrieved August 26, 2024, from https://openreview.net/forum?id=jpR98ZdIm2q (cit. on pp. 64, 102).

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., & Amodei, D. (2020, January 22). *Scaling Laws for Neural Language Models*. arXiv: 2001.08361 [cs, stat]. https://doi.org/10.48550/arXiv.2001.08361 (cit. on pp. 57, 69, 78, 128, 136).

Kaplun, G., Ghosh, N., Garg, S., Barak, B., & Nakkiran, P. (2022, February 20). *Deconstructing Distributions: A Pointwise Framework of Learning*. Retrieved April 5, 2022, from http://arxiv.org/abs/2202.09931 (cit. on pp. 90, 223).

Kasneci, E., Sessler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., Gasser, U., Groh, G., Günnemann, S., Hüllermeier, E., Krusche, S., Kutyniok, G., Michaeli, T., Nerdel, C., Pfeffer, J., Poquet, O., Sailer, M., Schmidt, A., Seidel, T., ... Kasneci, G. (2023). ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and Individual Differences*, *103*, 102274. https://doi.org/10.1016/j.lindif.2023.102274 (cit. on pp. 125, 144).

Kendall, A., & Gal, Y. (2017). What uncertainties do we need in Bayesian deep learning for computer vision? *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 5580–5590 (cit. on p. 18).

Kerschke, P., Hoos, H. H., Neumann, F., & Trautmann, H. (2019). Automated Algorithm Selection: Survey and Perspectives. *Evolutionary Computation*, *27*(1), 3–45. https://doi.org/10.1162/evco_a_00242 (cit. on p. 63).

Khanna, R., Dodge, J., Anderson, A., Dikkala, R., Irvine, J., Shureih, Z., Lam, K.-H., Matthews, C. R., Lin, Z., Kahng, M., Fern, A., & Burnett, M. (2022). Finding AI's Faults with AAR/AI: An Empirical Study. *ACM Transactions on Interactive Intelligent Systems*, *12*(1), 1:1–1:33. https://doi.org/10.1145/3487065 (cit. on p. 20).

Kim, G., Baldi, P., & McAleer, S. (2023, November 16). *Language Models can Solve Computer Tasks*. arXiv: 2303.17491 [cs]. https://doi.org/10.48550/arXiv.2303.17491 (cit. on p. 125).

Kim, T.-H., & Choi, J. (2018, June 6). *ScreenerNet: Learning Self-Paced Curriculum for Deep Neural Networks*. arXiv: 1801.00904 [cs]. https://doi.org/10.48550/arXiv.1801.00904 (cit. on p. 59).

Klein, E., Moeller, K., Dressel, K., Domahs, F., Wood, G., Willmes, K., & Nuerk, H.-C. (2010). To carry or not to carry—is this the question? disentangling the carry effect in multi-digit addition. *Acta psychologica*, *135*(1), 67–76 (cit. on p. 236).

Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., & Iwasawa, Y. (2022). Large language models are zero-shot reasoners. *Advances in neural information processing systems*, *35*, 22199–22213 (cit. on p. 260).

Kokiopoulou, E., Hauth, A., Sbaiz, L., Gesmundo, A., Bartók, G., & Berent, J. (2020). Task-Aware Performance Prediction for Efficient Architecture Search. In *ECAI 2020* (pp. 1238–1245). IOS Press. https://doi.org/10.3233/FAIA200224 (cit. on pp. 57, 81, 85).

Konda, V., & Tsitsiklis, J. (1999). Actor-Critic Algorithms. *Advances in Neural Information Processing Systems*, *12*. Retrieved September 30, 2024, from https://proceedings.neurips.cc/paper/1999/hash/6449f44a102fde848669bdd9eb6b76faAbstract.html (cit. on p. 59).

Könighofer, B., Rudolf, J., Palmisano, A., Tappler, M., & Bloem, R. (2023). Online shielding for reinforcement learning. *Innovations in Systems and Software Engineering*, *19*(4), 379–394. https://doi.org/10.1007/s11334-022-00480-4 (cit. on pp. 66, 91).

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix Factorization Techniques for Recommender Systems. *Computer*, *42*(8), 30–37. https://doi.org/10.1109/MC.2009.263 (cit. on p. 74).

Kossen, J., Farquhar, S., Gal, Y., & Rainforth, T. (2021). Active Testing: Sample-Efficient Model Evaluation. *Proceedings of the 38th International Conference on Machine Learning*, 5753–5763. Retrieved September 10, 2024, from https://proceedings.mlr.press/v139/kossen21a.html (cit. on p. 60).

Kossen, J., Farquhar, S., Gal, Y., & Rainforth, T. (2022, October 18). *Active Surrogate Estimators: An Active Learning Approach to Label-Efficient Model Evaluation*. arXiv: 2202.06881 [cs, stat]. https://doi.org/10.48550/arXiv.2202.06881 (cit. on pp. 60, 221).

Kuhn, L., Gal, Y., & Farquhar, S. (2022). Semantic Uncertainty: Linguistic Invariances for Uncertainty Estimation in Natural Language Generation. Retrieved

August 19, 2024, from https://openreview.net/forum?id=VD-AYtP0dve (cit. on pp. 127, 259).

Kuhn, T. S. (1996, December 15). *The Structure of Scientific Revolutions* (3rd edition). University of Chicago Press. (Cit. on pp. 2, 69).

Kukla, A. (1978). An attributional theory of choice. In *Advances in experimental social psychology* (pp. 113–144, Vol. 11). Elsevier. (Cit. on p. 261).

Kumar, S. (2022). Answer-level Calibration for Free-form Multiple Choice Question Answering. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 665–679. Retrieved May 30, 2022, from https://aclanthology.org/2022.acl-long.49 (cit. on p. 17).

Kuncheva, L. I. (2014, September 9). *Combining Pattern Classifiers: Methods and Algorithms*. John Wiley & Sons. (Cit. on pp. 15, 63, 222).

Lambert, N., Pyatkin, V., Morrison, J., Miranda, L., Lin, B. Y., Chandu, K., Dziri, N., Kumar, S., Zick, T., Choi, Y., Smith, N. A., & Hajishirzi, H. (2024). *RewardBench: Evaluating Reward Models for Language Modeling* (2). https://doi.org/10.48550/ARXIV.2403.13787 (cit. on pp. 91, 164).

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2020). ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. Retrieved December 16, 2022, from https://openreview.net/forum?id=H1eA7AEtvS (cit. on p. 110).

Learn and Fun. (2021). Unscramble word games: Anagram puzzle book (word scramble books for adults). *ISBN:9798443580959* (cit. on p. 228).

Lee, C.-H., Cheng, H., & Ostendorf, M. (2024, June). OrchestraLLM: Efficient Orchestration of Language Models for Dialogue State Tracking. 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 (Volume 1: Long Papers)* (pp. 1434–1445). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.naacl-long.79 (cit. on pp. 64, 86).

Lee, S., Wang, Z. J., Hoffman, J., & Chau, D. H. (2022, April 13). *VisCUIT: Visual Auditor for Bias in CNN Image Classifier*. Retrieved April 22, 2022, from http://arxiv.org/abs/2204.05899 (cit. on p. 20).

Li, C., & Flanigan, J. (2024). Task contamination: Language models may not be few-shot anymore. *Proceedings of the AAAI Conference on Artificial Intelligence*, *38*, 18471–18480 (cit. on p. 229).

Li, M., Wang, W., Feng, F., Zhu, F., Wang, Q., & Chua, T.-S. (2024, June 4). *Think Twice Before Trusting: Self-Detection for Large Language Models through Comprehensive Answer Reflection*. arXiv: 2403.09972 [cs]. https://doi.org/10.48550/arXiv.2403.09972 (cit. on pp. 141, 259).

Li, T., Chiang, W.-L., Frick, E., Dunlap, L., Wu, T., Zhu, B., Gonzalez, J. E., & Stoica, I. (2024, June 17). *From Crowdsourced Data to High-Quality Benchmarks: Arena-Hard and BenchBuilder Pipeline*. arXiv: 2406.11939 [cs]. https://doi.org/10.48550/arXiv.2406.11939 (cit. on p. 83).

Liang, P., Bommasani, R., Lee, T., Tsipras, D., Soylu, D., Yasunaga, M., Zhang, Y., Narayanan, D., Wu, Y., Kumar, A., Newman, B., Yuan, B., Yan, B., Zhang, C., Cosgrove, C., Manning, C. D., Ré, C., Acosta-Navas, D., Hudson, D. A., … Koreeda, Y. (2022, November 16). *Holistic Evaluation of Language Models*. arXiv: 2211.09110 [cs]. Retrieved November 17, 2022, from http://arxiv.org/abs/2211.09110 (cit. on pp. 48, 61, 80, 101, 124).

Lima, C. A. M., Coelho, A. L. V., & Von Zuben, F. J. (2007). Hybridizing mixtures of experts with support vector machines: Investigation into nonlinear dynamic systems identification. *Information Sciences*, *177*(10), 2049–2074. https://doi.org/10.1016/j.ins.2007.01.009 (cit. on p. 65).

Lin, S., Hilton, J., & Evans, O. (2022). Teaching Models to Express Their Uncertainty in Words. *Transactions on Machine Learning Research*. Retrieved January 20, 2023, from https://openreview.net/forum?id=8s8K2UZGTZ (cit. on p. 16).

Lipsey, R., Carlaw, K., & Bekar, C. (2006, January 1). *Economic Transformations: General Purpose Technologies and Long-Term Economic Growth*. Oxford University Press. https://doi.org/10.1093/oso/9780199285648.001.0001 (cit. on p. 13).

Liu, C., Zoph, B., Neumann, M., Shlens, J., Hua, W., Li, L.-J., Fei-Fei, L., Yuille, A., Huang, J., & Murphy, K. (2018). Progressive Neural Architecture Search. In V. Ferrari, M. Hebert, C. Sminchisescu, & Y. Weiss (Eds.), *Computer Vision – ECCV 2018* (pp. 19–35). Springer International Publishing. https://doi.org/10.1007/978-3-030-01246-5_2 (cit. on p. 57).

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2023). Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. *ACM Computing Surveys*, *55*(9), 195:1–195:35. https://doi.org/10.1145/3560815 (cit. on p. 126).

Liu, X., Shen, Y., Duh, K., & Gao, J. (2018, July). Stochastic Answer Networks for Machine Reading Comprehension. In I. Gurevych & Y. Miyao (Eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1694–1704). Association for Computational Linguistics. https://doi.org/10.18653/v1/P18-1157 (cit. on p. 20).

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. *ArXiv*. Retrieved December 16, 2022, from https://www.semanticscholar.org/paper/RoBERTa%3A-A-Robustly-Optimized-BERT-Pretraining-Liu-Ott/077f8329a7b6fa3b7c877a57b81eb6c18b5f87de (cit. on p. 110).

Llorca, D. F., Charisi, V., Hamon, R., Sánchez, I., & Gómez, E. (2023). Liability Regimes in the Age of AI: A Use-Case Driven Analysis of the Burden of Proof. *Journal of Artificial Intelligence Research*, *76*, 613–644. https://doi.org/10.1613/jair.1.14565 (cit. on p. 70).

Loecher, M., Lai, D., & Qi, W. (2022). Approximation of SHAP Values for Randomized Tree Ensembles. In A. Holzinger, P. Kieseberg, A. M. Tjoa, & E. Weippl (Eds.), *Machine Learning and Knowledge Extraction* (pp. 19–30). Springer International Publishing. https://doi.org/10.1007/978-3-031-14463-9_2 (cit. on p. 68).

Longino, H. E. (2002). *The Fate of Knowledge*. Princeton University Press. Retrieved June 27, 2024, from https://www.jstor.org/stable/j.ctv2tvzv0 (cit. on pp. 3, 53).

Lops, P., de Gemmis, M., & Semeraro, G. (2011). Content-based Recommender Systems: State of the Art and Trends. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), *Recommender Systems Handbook* (pp. 73–105). Springer US. https://doi.org/10.1007/978-0-387-85820-3_3 (cit. on p. 96).

Lorena, A. C., Paiva, P. Y. A., & Prudêncio, R. B. C. (2023). Trusting my predictions: On the value of Instance-Level analysis. *ACM Computing Surveys*, 3615354. https://doi.org/10.1145/3615354 (cit. on p. 223).

Lu, Y., Bartolo, M., Moore, A., Riedel, S., & Stenetorp, P. (2022). Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 8086–8098 (cit. on p. 259).

Ma, H., Zhang, C., Bian, Y., Liu, L., Zhang, Z., Zhao, P., Zhang, S., Fu, H., Hu, Q., & Wu, B. (2024). Fairness-guided few-shot prompting for large language models. *Advances in Neural Information Processing Systems*, *36* (cit. on p. 259).

Marecek, J., & Mettee, D. R. (1972). Avoidance of continued success as a function of self-esteem, level of esteem certainty, and responsibility for success. *Journal of Personality and Social Psychology*, *22*(1), 98–107 (cit. on pp. 128, 261).

Markou, M., & Singh, S. (2003). Novelty detection: A review—part 1: Statistical approaches. *Signal Processing*, *83*(12), 2481–2497. https://doi.org/10.1016/j.sigpro.2003.07.018 (cit. on pp. 18, 82).

Markov, T., Zhang, C., Agarwal, S., Nekoul, F. E., Lee, T., Adler, S., Jiang, A., & Weng, L. (2023). A holistic approach to undesired content detection in the real world. *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, *37*, 15009–15018. https://doi.org/10.1609/aaai.v37i12.26752 (cit. on p. 125).

Marshall, A. (2023). Robotaxis Can Now Work the Streets of San Francisco 24/7 [magazine]. *Wired*. Retrieved August 8, 2024, from https://www.wired.com/story/robotaxis-cruise-waymo-san-francisco/ (cit. on p. 1).

Martínez-Plumed, F., Prudêncio, R. B. C., Martínez-Usó, A., & Hernández-Orallo, J. (2016). Making sense of item response theory in machine learning. *Proceedings of the Twenty-second European Conference on Artificial Intelligence*, 1140–1148. https://doi.org/10.3233/978-1-61499-672-9-1140 (cit. on p. 77).

Martínez-Plumed, F., Prudêncio, R. B. C., Martínez-Usó, A., & Hernández-Orallo, J. (2019). Item response theory in AI: Analysing machine learning classifiers at the instance level. *Artificial Intelligence*, *271*, 18–42. https://doi.org/10.1016/j.artint.2018.09.004 (cit. on pp. 76, 77, 90).

Mayzner, M. S., & Tresselt, M. (1962). Anagram solution times: A function of word transition probabilities. *Journal of Experimental Psychology*, *63*(5), 510 (cit. on p. 228).

McCoy, R. T., Yao, S., Friedman, D., Hardy, M., & Griffiths, T. L. (2023). Embers of autoregression: Understanding large language models through the problem they are trained to solve. *arXiv preprint arXiv:2309.13638* (cit. on p. 260).

McGregor, M., Azzopardi, L., & Halvey, M. (2023). A systematic review of cost, effort, and load research in information search and retrieval, 1972–2020. *ACM Transactions on Information Systems*, *42*(1), 1–39 (cit. on p. 239).

McMullen, C., & Kivett, C. (2011). *Fun word scrambles for kids*. Independently published. https://books.google.es/books?id=9TeitgAACAAJ (cit. on p. 228).

Meding, K., Buschoff, L. M. S., Geirhos, R., & Wichmann, F. A. (2021). Trivial or Impossible — dichotomous data difficulty masks model differences (on ImageNet and beyond). Retrieved February 12, 2024, from https://openreview.net/forum?id=C_vsGwEIjAr (cit. on pp. 63, 89).

Meeds, E., & Osindero, S. (2005). An Alternative Infinite Mixture Of Gaussian Process Experts. *Advances in Neural Information Processing Systems*, *18*. Retrieved September 30, 2024, from https://papers.nips.cc/paper_files/paper/2005/hash/f499d34bd87b42948b3960b8f6b82e74-Abstract.html (cit. on p. 65).

Meehl, P. E. (1954). *Clinical versus statistical prediction: A theoretical analysis and a review of the evidence*. University of Minnesota Press. https://doi.org/10.1037/11281-000 (cit. on pp. 8, 224).

Meehl, P. E. (1967). Theory-Testing in Psychology and Physics: A Methodological Paradox. *Philosophy of Science*, *34*(2), 103–115. https://doi.org/10.1086/288135 (cit. on pp. 8, 224).

Menary, R. (2010). *The Extended Mind*. Cambridge, MA, USA: MIT Press. (Cit. on p. 42).

Meng, G., Zeng, Q., Lalor, J. P., & Yu, H. (2024, August 9). *A Psychology-based Unified Dynamic Framework for Curriculum Learning*. arXiv: 2408.05326 [cs]. Retrieved September 11, 2024, from http://arxiv.org/abs/2408.05326 (cit. on pp. 58, 77).

Merlin Bird ID. (2024). *Merlin Bird ID*. Retrieved August 8, 2024, from https://merlin.allaboutbirds.org (cit. on p. 1).

Merrick, L., & Taly, A. (2020). The Explanation Game: Explaining Machine Learning Models Using Shapley Values. In A. Holzinger, P. Kieseberg, A. M. Tjoa, & E. Weippl (Eds.), *Machine Learning and Knowledge Extraction* (pp. 17–38). Springer International Publishing. https://doi.org/10.1007/978-3-030-57321-8_2 (cit. on p. 67).

Messeri, L., & Crockett, M. J. (2024). Artificial intelligence and illusions of understanding in scientific research. *Nature*, *627*(8002), 49–58. https://doi.org/10.1038/s41586-024-07146-0 (cit. on p. 125).

Mielke, S. J., Szlam, A., Dinan, E., & Boureau, Y.-L. (2022). Reducing Conversational Agents' Overconfidence Through Linguistic Calibration (B. Roark & A. Nenkova, Eds.). *Transactions of the Association for Computational Linguistics*, *10*, 857–872. https://doi.org/10.1162/tacl_a_00494 (cit. on p. 16).

Mihaylov, T., Clark, P., Khot, T., & Sabharwal, A. (2018, October). Can a Suit of Armor Conduct Electricity? A New Dataset for Open Book Question Answering. In E. Riloff, D. Chiang, J. Hockenmaier, & J. Tsujii (Eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp. 2381–2391). Association for Computational Linguistics. https://doi.org/10.18653/v1/D18-1260 (cit. on pp. 128, 146).

Miller, J. P., Taori, R., Raghunathan, A., Sagawa, S., Koh, P. W., Shankar, V., Liang, P., Carmon, Y., & Schmidt, L. (2021). Accuracy on the Line: On the Strong Correlation Between Out-of-Distribution and In-Distribution

Generalization. *Proceedings of the 38th International Conference on Machine Learning*, 7721–7735. Retrieved September 11, 2024, from https://proceedings.mlr.press/v139/miller21b.html (cit. on p. 82).

Mishra, S., Khashabi, D., Baral, C., & Hajishirzi, H. (2022). Cross-Task Generalization via Natural Language Crowdsourcing Instructions. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 3470–3487. https://doi.org/10.18653/v1/2022.acl-long.244 (cit. on p. 146).

Mitchell, T. M. (1997). *Machine learning* (Nachdr., Vol. 1). McGraw-Hill. (Cit. on p. 72).

Morgan, C. (2021). *Anagrams 5-letter vocabulary building word puzzles and other games: Education resources by bounce learning kids*. Independently Published. https://books.google.es/books?id=QQC9zgEACAAJ (cit. on p. 228).

Mueller, S., Li, A., & Pearl, J. (2021, May 2). *Causes of Effects: Learning individual responses from population data*. Retrieved May 4, 2021, from http://arxiv.org/abs/2104.13730 (cit. on p. 223).

Muennighoff, N., Wang, T., Sutawika, L., Roberts, A., Biderman, S., Le Scao, T., Bari, M. S., Shen, S., Yong, Z. X., Schoelkopf, H., Tang, X., Radev, D., Aji, A. F., Almubarak, K., Albanie, S., Alyafeai, Z., Webson, A., Raff, E., & Raffel, C. (2023). Crosslingual Generalization through Multitask Finetuning. *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 15991–16111. https://doi.org/10.18653/v1/2023.acl-long.891 (cit. on pp. 128, 148, 227).

Mulaik, S. A. (2009, September 25). *Foundations of Factor Analysis* (2nd edition). Chapman and Hall/CRC. (Cit. on p. 74).

Muñoz, M. A., Villanova, L., Baatar, D., & Smith-Miles, K. (2018). Instance spaces for machine learning classification. *Machine Learning*, *107*(1), 109–147. https://doi.org/10.1007/s10994-017-5629-5 (cit. on p. 91).

Murphy, K. P. (2022). *Probabilistic machine learning: An introduction*. The MIT Press. probml.ai (cit. on p. 15).

Murphy, K. P. (2023). *Probabilistic machine learning: Advanced topics*. The MIT Press. (Cit. on pp. 15, 18, 47, 88).

Murray, K., & Chiang, D. (2018). Correcting Length Bias in Neural Machine Translation. *Proceedings of the Third Conference on Machine Translation: Research Papers*, 212–223. https://doi.org/10.18653/v1/W18-6322 (cit. on p. 16).

Nadeem, M. S. A., Zucker, J.-D., & Hanczar, B. (2009). Accuracy-Rejection Curves (ARCs) for Comparing Classification Methods with a Reject Option. *Proceedings of the Third International Workshop on Machine Learning in Systems Biology*, 65–81. Retrieved August 13, 2024, from https://proceedings.mlr.press/v8/nadeem10a.html (cit. on p. 41).

NannyML. (2024). *Nanny Models: Estimation of Performance of the Monitored Model*. NannyML. Retrieved May 31, 2024, from https://web.archive.org/web/20240822213439/https://nannyml.readthedocs.io/en/stable/how_it_works/performance_estimation.html#direct-loss-estimation-dle (cit. on pp. 35, 66, 100, 105, 155, 156).

Nguyen, Q. H., Hoang, D. C., Decugis, J., Manchanda, S., Chawla, N. V., & Doan, K. D. (2024). MetaLLM: A High-performant and Cost-efficient Dynamic Framework for Wrapping LLMs. https://doi.org/10.48550/ARXIV.2407.10834 (cit. on p. 64).

Ni, J., Xue, F., Yue, X., Deng, Y., Shah, M., Jain, K., Neubig, G., & You, Y. (2024, June 3). *MixEval: Deriving Wisdom of the Crowd from LLM Benchmark Mixtures*. arXiv: 2406.06565 [cs]. https://doi.org/10.48550/arXiv.2406.06565 (cit. on p. 61).

Nicholls, J. G. (1984). Achievement motivation: Conceptions of ability, subjective experience, task choice, and performance. *Psychological review*, *91*(3), 328 (cit. on p. 261).

Nixon, J., Dusenberry, M. W., Zhang, L., Jerfel, G., & Tran, D. (2019). Measuring Calibration in Deep Learning, 4. https://openaccess.thecvf.com/content_CVPRW_2019/html/Uncertainty_and_Robustness_in_Deep_Visual_Learning/Nixon_Measuring_Calibration_in_Deep_Learning_CVPRW_2019_paper.html (cit. on p. 17).

Ong, I., Almahairi, A., Wu, V., Chiang, W.-L., Wu, T., Gonzalez, J. E., Kadous, M. W., & Stoica, I. (2024, July 1). *RouteLLM: Learning to Route LLMs with Preference Data*. arXiv: 2406.18665 [cs]. https://doi.org/10.48550/arXiv.2406.18665 (cit. on pp. 34, 64, 86, 102).

OpenAI. (2023a). Gpt-3.5 turbo fine-tuning and api updates [Accessed: 2023-09-17]. *https://openai.com/blog/gpt-3-5-turbo-fine-tuning-and-api-updates* (cit. on p. 227).

OpenAI. (2023b). *ModelIndex*. Retrieved August 19, 2024, from https://platform.openai.com (cit. on p. 130).

OpenAI. (2023c, March 27). *GPT-4 Technical Report*. arXiv: 2303.08774 [cs]. https://doi.org/10.48550/arXiv.2303.08774 (cit. on pp. 125, 128, 141–143, 148, 227).

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Gray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., & Lowe, R. (2022). Training language models to follow instructions with human feedback. Retrieved June 1, 2023, from https://openreview.net/forum?id=TG8KACxEON (cit. on pp. 7, 14, 66, 125, 148, 227).

Owen, D. (2024, January 9). *How predictable is language model benchmark performance?* arXiv: 2401.04757 [cs]. Retrieved January 15, 2024, from http://arxiv.org/abs/2401.04757 (cit. on pp. 58, 78).

Pacchiardi, L., Cheke, L., & Hernández-Orallo, J. (2024). 100 instances is all you need: Predicting the success of a new LLM on unseen data by testing on a few instances. https://genai-evaluation-kdd2024.github.io/genai-evalution-kdd2024/assets/papers/GenAI_Evaluation_KDD2024_paper_18.pdf (cit. on p. 61).

Pan, A., Chan, J. S., Zou, A., Li, N., Basart, S., Woodside, T., Zhang, H., Emmons, S., & Hendrycks, D. (2023). Do the rewards justify the means? measuring trade-offs between rewards and ethical behavior in the machiavelli benchmark. *International Conference on Machine Learning*, 26837–26867 (cit. on p. 235).

Panda, D. K., & Ray, S. (2022). Approaches and algorithms to mitigate cold start problems in recommender systems: A systematic literature review. *Journal of Intelligent Information Systems*, *59*(2), 341–366. https://doi.org/10.1007/s10844-022-00698-5 (cit. on p. 96).

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). BLEU: A method for automatic evaluation of machine translation. *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, 311–318. https://doi.org/10.3115/1073083.1073135 (cit. on p. 19).

Park, H., & Stefanski, L. A. (1998). Relative-error prediction. *Statistics & Probability Letters*, *40*(3), 227–236. https://doi.org/10.1016/S0167-7152(98)00088-1 (cit. on p. 101).

Parker-Holder, J., Jiang, M., Dennis, M., Samvelyan, M., Foerster, J., Grefenstette, E., & Rocktäschel, T. (2022, March 2). *Evolving Curricula with Regret-Based Environment Design*. Retrieved March 3, 2022, from http://arxiv.org/abs/2203.01302 (cit. on p. 59).

Platt, J. (1999). Probabilistic Outputs for Support vector Machines and Comparisons to Regularized Likelihood Methods. In *Advances in large margin classifiers* (pp. 61–74, Vol. 10). MIT Press. Retrieved August 21, 2024, from https://www.semanticscholar.org/paper/Probabilistic-Outputs-for-Support-vector-Machines-Platt/42e5ed832d4310ce4378c44d05570439df28a393 (cit. on pp. 15, 17, 253).

Polo, F. M., Weber, L., Choshen, L., Sun, Y., Xu, G., & Yurochkin, M. (2024, February 22). *tinyBenchmarks: Evaluating LLMs with fewer examples*. arXiv: 2402.14992 [cs, stat]. https://doi.org/10.48550/arXiv.2402.14992 (cit. on pp. 61, 77).

Popper, K. R. (1959). *The Logic of Scientific Discovery*. Basic Books. (Cit. on p. 2).

Premakumar, V. N., Vaiana, M., Pop, F., Rosenblatt, J., de Lucena, D. S., Ziman, K., & Graziano, M. S. A. (2024, July 14). *Unexpected Benefits of Self-Modeling in Neural Systems*. arXiv: 2407.10188 [cs]. Retrieved July 21, 2024, from http://arxiv.org/abs/2407.10188 (cit. on p. 165).

Prudêncio, R. B. C., Lorena, A. C., Silva-Filho, T., Drapal, P., & Valeriano, M. G. (2024). Assessor Models for Explaining Instance Hardness in Classification Problems. *2024 International Joint Conference on Neural Networks (IJCNN)*, 1–8. https://doi.org/10.1109/IJCNN60899.2024.10651521 (cit. on pp. 68, 86).

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). *Improving Language Understanding by Generative Pre-Training*. OpenAI. (Cit. on p. 148).

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language Models are Unsupervised Multitask Learners (cit. on p. 110).

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, *21*(140), 1–67. Retrieved May 30, 2022, from http://jmlr.org/papers/v21/20-074.html (cit. on p. 125).

Raimondi, R., Tzoumas, N., Salisbury, T., Di Simplicio, S., & Romano, M. R. (2023). Comparative analysis of large language models in the royal college of ophthalmologists fellowship exams. *Eye*, *37*(17), 3530–3533 (cit. on p. 259).

Rajpurkar, P., Jia, R., & Liang, P. (2018). Know what you don't know: Unanswerable questions for squad. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 784–789 (cit. on p. 228).

Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). Squad: 100,000+ questions for machine comprehension of text. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (cit. on p. 228).

Rein, D., Hou, B. L., Stickland, A. C., Petty, J., Pang, R. Y., Dirani, J., Michael, J., & Bowman, S. R. (2023, November 20). *GPQA: A Graduate-Level Google-Proof Q&A Benchmark*. arXiv: 2311.12022 [cs]. https://doi.org/10.48550/arXiv.2311.12022 (cit. on pp. 144, 146).

Rescher, N. (1958). On prediction and explanation. *The British Journal for the Philosophy of Science*, *8*(32), 281–290. https://doi.org/10.1093/bjps/VIII.32.281 (cit. on p. 3).

Ribeiro, M. T., Singh, S., & Guestrin, C. (2018, July). Semantically Equivalent Adversarial Rules for Debugging NLP models. In I. Gurevych & Y. Miyao (Eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 856–865). Association for Computational Linguistics. https://doi.org/10.18653/v1/P18-1079 (cit. on p. 20).

Rice, J. R. (1976). The Algorithm Selection Problem. In *Advances in Computers* (pp. 65–118, Vol. 15). Elsevier. Retrieved February 23, 2022, from https://www.sciencedirect.com/science/article/pii/S0065245808605203 (cit. on p. 63).

Risi, S., & Togelius, J. (2020). Increasing generality in machine learning through procedural content generation. *Nature Machine Intelligence*, *2*(8), 428–436. https://doi.org/10.1038/s42256-020-0208-z (cit. on p. 59).

Ruan, Y., Maddison, C. J., & Hashimoto, T. (2024, May 17). *Observational Scaling Laws and the Predictability of Language Model Performance*. arXiv: 2405.10938 [cs, stat]. https://doi.org/10.48550/arXiv.2405.10938 (cit. on p. 89).

Rutar, D., Schellaert, W., Markelius, A., Cheke, L. G., & Hernández-Orallo, J. (2024, June 19). *General Interaction Battery: Simple Object Navigation and Affordances (Gibsona)*. https://doi.org/10.2139/ssrn.4871025 (cit. on pp. 88, 219).

Sahoo, P., Singh, A. K., Saha, S., Jain, V., Mondal, S., & Chadha, A. (2024). *A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications* (1). https://doi.org/10.48550/ARXIV.2402.07927 (cit. on p. 42).

Salewski, L., Alaniz, S., Rio-Torto, I., Schulz, E., & Akata, Z. (2023). In-Context Impersonation Reveals Large Language Models' Strengths and Biases. *Advances in Neural Information Processing Systems*, *36*, 72044–72057. Retrieved September 5, 2024, from https://proceedings.neurips.cc/paper_files/paper/2023/hash/e3fe7b34ba4f378df39cb12a97193f41-Abstract-Conference.html (cit. on p. 154).

Sanh, V., Webson, A., Raffel, C., Bach, S., Sutawika, L., Alyafeai, Z., Chaffin, A., Stiegler, A., Raja, A., Dey, M., Bari, M. S., Xu, C., Thakker, U., Sharma,

S. S., Szczechla, E., Kim, T., Chhablani, G., Nayak, N., Datta, D., ... Rush, A. M. (2021). Multitask Prompted Training Enables Zero-Shot Task Generalization. Retrieved August 19, 2024, from https://openreview.net/forum?id=9Vrb9D0WI4 (cit. on pp. 126, 127, 259).

Saunders, W., Yeh, C., Wu, J., Bills, S., Ouyang, L., Ward, J., & Leike, J. (2022, June 13). *Self-critiquing models for assisting human evaluators*. arXiv: 2206.05802 [cs]. Retrieved August 27, 2024, from http://arxiv.org/abs/2206.05802 (cit. on p. 224).

Schaeffer, R., Miranda, B., & Koyejo, S. (2023). Are Emergent Abilities of Large Language Models a Mirage? *Deployable Generative AI Workshop at ICML*. Retrieved November 3, 2023, from https://openreview.net/forum?id=JRdN9GcI52 (cit. on pp. 78, 100).

Schaeffer, R., Schoelkopf, H., Miranda, B., Mukobi, G., Madan, V., Ibrahim, A., Bradley, H., Biderman, S., & Koyejo, S. (2024, June 6). *Why Has Predicting Downstream Capabilities of Frontier AI Models with Scale Remained Elusive?* arXiv: 2406.04391 [cs]. Retrieved June 12, 2024, from http://arxiv.org/abs/2406.04391 (cit. on pp. 69, 78, 100, 101, 226).

Schaul, K., Chen, S. Y., & Tiku, N. (2023). Inside the secret list of websites that make AI like ChatGPT sound smart [newspaper]. *Washington Post*. Retrieved August 19, 2024, from https://www.washingtonpost.com/technology/interactive/2023/ai-chatbot-learning/ (cit. on p. 142).

Schellaert, W., Hamon, R., Martínez-Plumed, F., & Hernandez-Orallo, J. (2024, March). A Proposal for Scaling the Scaling Laws. In A. V. Miceli-Barone, F. Barez, S. Cohen, E. Voita, U. Germann, & M. Lukasik (Eds.), *Proceedings of the First edition of the Workshop on the Scaling Behavior of Large Language Models (SCALE-LLM 2024)* (pp. 1–8). Association for Computational Linguistics. Retrieved June 17, 2024, from https://aclanthology.org/2024.scalellm-1.1 (cit. on pp. 7, 78, 123, 218).

Schellaert, W., Martínez-Plumed, F., & Hernández-Orallo, J. (2024). Analysing the Predictability of Language Model Performance. *ACM Trans. Intell. Syst. Technol.* https://doi.org/10.1145/3706118 (cit. on p. 219).

Schellaert, W., Martínez-Plumed, F., Vold, K., Burden, J., Casares, P. A. M., Loe, B. S., Reichart, R., hÉigeartaigh, S. Ó., Korhonen, A., & Hernández-Orallo,

J. (2023). Your Prompt is My Command: On Assessing the Human-Centred Generality of Multimodal Models. *Journal of Artificial Intelligence Research*, *77*, 377–394. https://doi.org/10.1613/jair.1.14157 (cit. on pp. 7, 42, 126, 218).

Schelter, S., Rukat, T., & Biessmann, F. (2020). Learning to Validate the Predictions of Black Box Classifiers on Unseen Data. *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, 1289–1299. https://doi.org/10.1145/3318464.3380604 (cit. on p. 62).

Schram, V., Beck, D., & Cohn, T. (2023, May). Performance Prediction via Bayesian Matrix Factorisation for Multilingual Natural Language Processing Tasks. In A. Vlachos & I. Augenstein (Eds.), *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics* (pp. 1790–1801). Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.eacl-main.131 (cit. on p. 75).

Sclar, M., Choi, Y., Tsvetkov, Y., & Suhr, A. (2023). Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. *The Twelfth International Conference on Learning Representations* (cit. on p. 259).

Sebeok, T. A. (2001). *Signs: An introduction to semiotics*. University of Toronto Press. (Cit. on p. 229).

Seeger, M. (2004). Gaussian processes for machine learning. *International Journal of Neural Systems*, *14* (02), 69–106. https://doi.org/10.1142/S0129065704001899 (cit. on p. 81).

Settles, B. (2010, January 17). *Active Learning Literature Survey*. Retrieved February 24, 2022, from http://burrsettles.com/pub/settles.activelearning.pdf (cit. on p. 59).

Shannon, C. E. (1948). A Mathematical Theory of Communication. *The Bell System Technical Journal*, *27*(3), 379–423. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x (cit. on p. 14).

Shavit, Y., O'Keefe, C., Eloundou, T., McMillan, P., Agarwal, S., Brundage, M., Adler, S., Campbell, R., Lee, T., Mishkin, P., Hickey, A., Slama, K., Ahmad, L., Beutel, A., Passos, A., & Robinson, D. G. (2023). *Practices for Governing*

Agentic AI Systems. OpenAI. Retrieved December 9, 2024, from https://cdn.openai.com/papers/practices-for-governing-agentic-ai-systems.pdf (cit. on p. 65).

Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., & Dean, J. (2017). Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. *ICLR*. Retrieved September 17, 2021, from https://openreview.net/forum?id=B1ckMDqlg (cit. on p. 66).

She, J. S., Potts, C., Bowman, S. R., & Geiger, A. (2023, July). ScoNe: Benchmarking Negation Reasoning in Language Models With Fine-Tuning and In-Context Learning. In A. Rogers, J. Boyd-Graber, & N. Okazaki (Eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (pp. 1803–1821). Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.acl-short.154 (cit. on pp. 3, 20).

Shi, Z., & Lipani, A. (2023). Don't stop pretraining? make prompt-based fine-tuning powerful learner. *Advances in Neural Information Processing Systems*, *36*, 5827–5849 (cit. on p. 259).

Shmueli, G. (2010). To Explain or to Predict? *Statistical Science*, *25*(3). https://doi.org/10.1214/10-STS330 (cit. on pp. 5, 161).

Shnitzer, T., Ou, A., Silva, M., Soule, K., Sun, Y., Solomon, J., Thompson, N., & Yurochkin, M. (2023, September 27). *Large Language Model Routing with Benchmark Datasets*. arXiv: 2309.15789 [cs]. https://doi.org/10.48550/arXiv.2309.15789 (cit. on pp. 34, 64, 85, 86, 105, 155, 156).

Siddiqui, S. A., Rajkumar, N., Maharaj, T., Krueger, D., & Hooker, S. (2022, September 20). *Metadata Archaeology: Unearthing Data Subsets by Leveraging Training Dynamics*. arXiv: 2209.10015 [cs]. https://doi.org/10.48550/arXiv.2209.10015 (cit. on p. 80).

SimpleMaps. (2023). *World Cities Database*. Retrieved August 19, 2024, from https://simplemaps.com/data/world-cities (cit. on p. 146).

Sinha, A., Malo, P., & Deb, K. (2020, December 5). *A Review on Bilevel Optimization: From Classical to Evolutionary Approaches and Applications*. arXiv:

1705.06270 [cs, math]. Retrieved September 8, 2024, from http://arxiv.org/abs/1705.06270 (cit. on p. 166).

Song, H., Diethe, T., Kull, M., & Flach, P. (2019). Distribution calibration for regression. *Proceedings of the 36th International Conference on Machine Learning*, 5897–5906. Retrieved September 20, 2024, from https://proceedings.mlr.press/v97/song19a.html (cit. on p. 17).

Song, X., Li, O., Lee, C., Yang, B., Peng, D., Perel, S., & Chen, Y. (2024, March 4). *OmniPred: Language Models as Universal Regressors*. arXiv: 2402.14547 [cs]. https://doi.org/10.48550/arXiv.2402.14547 (cit. on pp. 85, 102).

Song, X., Zhang, Q., Lee, C., Fertig, E., Huang, T.-K., Belenki, L., Kochanski, G., Ariafar, S., Vasudevan, S., Perel, S., & Golovin, D. (2024, August 21). *The Vizier Gaussian Process Bandit Algorithm*. arXiv: 2408.11527 [cs, math]. https://doi.org/10.48550/arXiv.2408.11527 (cit. on p. 55).

Soviany, P., Ionescu, R. T., Rota, P., & Sebe, N. (2022). Curriculum Learning: A Survey. *International Journal of Computer Vision*, *130*(6), 1526–1565. https://doi.org/10.1007/s11263-022-01611-x (cit. on pp. 58, 83).

Srinivas, K., & Roediger III, H. L. (1990). Classifying implicit memory tests: Category association and anagram solution. *Journal of Memory and Language*, *29*(4), 389–412 (cit. on p. 228).

Srinivasan, A., Kholkar, G., Kejriwal, R., Ganu, T., Dandapat, S., Sitaram, S., Santhanam, B., Aditya, S., Bali, K., & Choudhury, M. (2022). LITMUS Predictor: An AI Assistant for Building Reliable, High-Performing and Fair Multilingual NLP Systems. *AAAI*. Retrieved February 25, 2022, from https://aaai-2022.virtualchair.net/poster_dm144 (cit. on p. 85).

Srivastava, A., Rastogi, A., Rao, A., Shoeb, A. A. M., Abid, A., Fisch, A., Brown, A. R., Santoro, A., Gupta, A., Garriga-Alonso, A., et al. (2023). Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Journal of Machine Learning Research* (cit. on p. 262).

Srivastava, A., Rastogi, A., Rao, A., Shoeb, A. A. M., Abid, A., Fisch, A., Brown, A. R., Santoro, A., Gupta, A., Garriga-Alonso, A., Kluska, A., Lewkowycz, A., Agarwal, A., Power, A., Ray, A., Warstadt, A., Kocurek, A. W., Safaya, A., Tazarv, A., … Wu, Z. (2022). *Beyond the Imitation Game: Quantifying*

and extrapolating the capabilities of language models. arXiv: 2206.04615 [cs, stat]. https://doi.org/10.48550/arXiv.2206.04615 (cit. on pp. 61, 80, 101, 106, 108, 109).

Staneva, M., Baret, A., Aso-Mollar, Á., Blass, J., Ponz, S. C., Conitzer, V., Cortes, U., Dasigi, P., Paula, A. de, Galindo, C., Gobert, J., Gonzàlez, J., Heintz, F., Hendler, J., Hendrycks, D., Hunter, L., Izquierdo-Domenech, J., Juarez, M., Frias, A. J., ... Wooldridge, M. (2023, November 16). *Assessing AI capabilities with education tests*. OECD. Paris. https://doi.org/10.1787/bbdeb1e0-en (cit. on p. 218).

SB 1047 (n.d.). Retrieved September 29, 2024, from https://leginfo.legislature.ca.gov/faces/billNavClient.xhtml?bill_id=202320240SB1047 (cit. on p. 7).

Steyvers, M., Tejeda, H., Kumar, A., Belem, C., Karny, S., Hu, X., Mayer, L., & Smyth, P. (2024, January 24). *The Calibration Gap between Model and Human Confidence in Large Language Models*. arXiv: 2401.13835 [cs]. Retrieved January 30, 2024, from http://arxiv.org/abs/2401.13835 (cit. on pp. 4, 17, 26).

Streeter, M. (2018). Approximation Algorithms for Cascading Prediction Models. *Proceedings of the 35th International Conference on Machine Learning*, 4752–4760. Retrieved February 12, 2024, from https://proceedings.mlr.press/v80/streeter18a.html (cit. on p. 63).

Sun, J., Shaib, C., & Wallace, B. C. (2023, July 8). *Evaluating the Zero-shot Robustness of Instruction-tuned Language Models*. arXiv: 2306.11270 [cs]. Retrieved December 23, 2023, from http://arxiv.org/abs/2306.11270 (cit. on pp. 126, 127, 228, 259).

Sun, Y., Wang, H., Xue, B., Jin, Y., Yen, G. G., & Zhang, M. (2020). Surrogate-Assisted Evolutionary Deep Learning Using an End-to-End Random Forest-Based Performance Predictor. *IEEE Transactions on Evolutionary Computation*, *24*(2), 350–364. https://doi.org/10.1109/TEVC.2019.2924461 (cit. on pp. 56, 81).

Swayamdipta, S., Schwartz, R., Lourie, N., Wang, Y., Hajishirzi, H., Smith, N. A., & Choi, Y. (2020). Dataset Cartography: Mapping and Diagnosing Datasets with Training Dynamics. *Proceedings of the 2020 Conference on Empirical*

Methods in Natural Language Processing (EMNLP), 9275–9293. https://doi.org/10.18653/v1/2020.emnlp-main.746 (cit. on pp. 74, 80, 164, 223).

Tal, E. (2020). Measurement in Science. In E. N. Zalta (Ed.), *The Stanford Encyclopedia of Philosophy* (Fall 2020). Metaphysics Research Lab, Stanford University. Retrieved February 16, 2024, from https://plato.stanford.edu/archives/fall2020/entries/measurement-science/ (cit. on p. 160).

Tang, Y., Wang, Y., Xu, Y., Chen, H., Shi, B., Xu, C., Xu, C., Tian, Q., & Xu, C. (2020). A Semi-Supervised Assessor of Neural Architectures. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 1807–1816. https://doi.org/10.1109/CVPR42600.2020.00188 (cit. on p. 81).

Thirunavukarasu, A. J., Ting, D. S. J., Elangovan, K., Gutierrez, L., Tan, T. F., & Ting, D. S. W. (2023). Large language models in medicine. *Nature Medicine*, *29*(8), 1930–1940. https://doi.org/10.1038/s41591-023-02448-8 (cit. on p. 125).

Thoppilan, R., De Freitas, D., Hall, J., Shazeer, N., Kulshreshtha, A., Cheng, H.-T., Jin, A., Bos, T., Baker, L., Du, Y., Li, Y., Lee, H., Zheng, H. S., Ghafouri, A., Menegali, M., Huang, Y., Krikun, M., Lepikhin, D., Qin, J., ... Le, Q. (2022, February 10). *LaMDA: Language Models for Dialog Applications*. arXiv: 2201.08239 [cs]. https://doi.org/10.48550/arXiv.2201.08239 (cit. on p. 108).

Thorndyke, P. W. (1979, January 1). *Distance estimation from cognitive maps*. RAND Corporation. Retrieved August 19, 2024, from https://www.rand.org/pubs/reports/R2474.html (cit. on p. 144).

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., & Lample, G. (2023, February 27). *LLaMA: Open and Efficient Foundation Language Models*. arXiv: 2302.13971 [cs]. Retrieved September 25, 2023, from http://arxiv.org/abs/2302.13971 (cit. on pp. 69, 148, 227).

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., ... Scialom, T. (2023, July 19). *Llama 2: Open Foundation and Fine-Tuned Chat*

Models. arXiv: 2307.09288 [cs]. https://doi.org/10.48550/arXiv.2307.09288 (cit. on pp. 69, 127, 148, 227).

Tsigaris, P., & Teixeira da Silva, J. A. (2023). Can ChatGPT be trusted to provide reliable estimates? *Accountability in Research*, *0*(0), 1–3. https://doi.org/10.1080/08989621.2023.2179919 (cit. on p. 126).

Udandarao, V., Prabhu, A., Ghosh, A., Sharma, Y., Torr, P. H. S., Bibi, A., Albanie, S., & Bethge, M. (2024, April 8). *No "Zero-Shot" Without Exponential Data: Pretraining Concept Frequency Determines Multimodal Model Performance*. arXiv: 2404.04125 [cs]. https://doi.org/10.48550/arXiv.2404.04125 (cit. on p. 89).

Varshney, N., Mishra, S., & Baral, C. (2022). ILDAE: Instance-Level Difficulty Analysis of Evaluation Data. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 3412–3425. https://doi.org/10.18653/v1/2022.acl-long.240 (cit. on p. 73).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is All you Need. *Advances in Neural Information Processing Systems*, *30*. Retrieved January 11, 2023, from https://papers.nips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html (cit. on p. 14).

Vilalta, R., & Drissi, Y. (2002). A Perspective View and Survey of Meta-Learning. *Artificial Intelligence Review*, *18*(2), 77–95. https://doi.org/10.1023/A:1019956318069 (cit. on pp. 50, 84, 163).

Villalobos, P., Sevilla, J., Heim, L., Besiroglu, T., Hobbhahn, M., & Ho, A. (2022, October 25). *Will we run out of data? An analysis of the limits of scaling datasets in Machine Learning*. arXiv: 2211.04325 [cs]. https://doi.org/10.48550/arXiv.2211.04325 (cit. on p. 142).

Voudouris, K., Alhas, I., Schellaert, W., Crosby, M., Holmes, J., Burden, J., Chaubey, N., Donnelly, N., Patel, M., Halina, M., Hernández-Orallo, J., & Cheke, L. G. (2023, December 18). *Animal-AI 3: What's New & Why You Should Care*. Retrieved January 10, 2024, from https://arxiv.org/abs/2312.11414v1 (cit. on pp. 7, 88, 218, 219).

Voudouris, K., Liu, J. D., Siwinska, N., Schellaert, W., & Cheke, L. G. (2024). Investigating Object Permanence in Deep Reinforcement Learning Agents. *Proceedings of the Annual Meeting of the Cognitive Science Society*, *46*(0). Retrieved June 17, 2024, from https://escholarship.org/uc/item/3g6575z9 (cit. on pp. 7, 88, 219).

Waga, M., Castellano, E., Pruekprasert, S., Klikovits, S., Takisaka, T., & Hasuo, I. (2022). Dynamic Shielding for Reinforcement Learning in Black-Box Environments. In A. Bouajjani, L. Holík, & Z. Wu (Eds.), *Automated Technology for Verification and Analysis* (pp. 25–41). Springer International Publishing. https://doi.org/10.1007/978-3-031-19992-9_2 (cit. on pp. 66, 91, 104).

Walker, M. P., Liston, C., Hobson, J. A., & Stickgold, R. (2002). Cognitive flexibility across the sleep–wake cycle: Rem-sleep enhancement of anagram problem solving. *Cognitive Brain Research*, *14*(3), 317–324 (cit. on p. 228).

Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. (2019). SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems. *Advances in Neural Information Processing Systems*, *32*. Retrieved September 7, 2021, from https://papers.nips.cc/paper/2019/hash/4496bf24afe7fab6f046bf4923da8de6-Abstract.html (cit. on pp. 30, 110).

Wang, H., Li, J., Wu, H., Hovy, E., & Sun, Y. (2023). Pre-Trained Language Models and Their Applications. *Engineering*, *25*, 51–65. https://doi.org/10.1016/j.eng.2022.04.024 (cit. on p. 109).

Wang, Y., Mishra, S., Alipoormolabashi, P., Kordi, Y., Mirzaei, A., Naik, A., Ashok, A., Dhanasekaran, A. S., Arunkumar, A., Stap, D., Pathak, E., Karamanolakis, G., Lai, H., Purohit, I., Mondal, I., Anderson, J., Kuznia, K., Doshi, K., Pal, K. K., … Shen, X. (2022, December). Super-NaturalInstructions: Generalization via Declarative Instructions on 1600+ NLP Tasks. In Y. Goldberg, Z. Kozareva, & Y. Zhang (Eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing* (pp. 5085–5109). Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.emnlp-main.340 (cit. on pp. 126, 127, 259).

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., Chi, E. H., Hashimoto, T., Vinyals, O., Liang, P., Dean, J., & Fedus, W. (2022). Emergent Abilities of Large

Language Models. *Transactions on Machine Learning Research*. Retrieved November 15, 2022, from https://openreview.net/forum?id=yzkSU5zdwD (cit. on p. 78).

Westerlund, M. (2019). The Emergence of Deepfake Technology: A Review. *Technology Innovation Management Review*, *9*(11), 40–53. https://doi.org/10.22215/timreview/1282 (cit. on p. 1).

Whewell, W. (1969). Mr. Mill's Logic. In R. Butts (Ed.), *William Whewell's Theory of Scientific Method (Republished 1969)* (1st edition). University of Pittsburgh Press. (Original work published 1849). (Cit. on pp. 2, 53, 158)

Witte, K. L., & Freund, J. S. (1995). Anagram solution as related to adult age, anagram difficulty, and experience in solving crossword puzzles. *Aging, Neuropsychology, and Cognition*, *2*(2), 146–155 (cit. on p. 237).

Witte, K. L., Freund, J. S., & Csiki, I. (2002a). Case-mixing effects on anagram solution. *The Journal of general psychology*, *129*(2), 117–126 (cit. on p. 228).

Witte, K. L., Freund, J. S., & Csiki, I. (2002b). Case-Mixing Effects on Anagram Solution. *The Journal of General Psychology*, *129*(2), 117–126. https://doi.org/10.1080/00221300209603132 (cit. on p. 144).

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., ... Rush, A. (2020). Transformers: State-of-the-Art Natural Language Processing. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 38–45. https://doi.org/10.18653/v1/2020.emnlp-demos.6 (cit. on p. 110).

Wu, T., Ribeiro, M. T., Heer, J., & Weld, D. (2019, July). Errudite: Scalable, Reproducible, and Testable Error Analysis. In A. Korhonen, D. Traum, & L. Màrquez (Eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 747–763). Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-1073 (cit. on p. 20).

Xie, T., Zhang, D., Chen, J., Li, X., Zhao, S., Cao, R., Hua, T. J., Cheng, Z., Shin, D., Lei, F., Liu, Y., Xu, Y., Zhou, S., Savarese, S., Xiong, C., Zhong, V., & Yu, T. (2024, April 11). *OSWorld: Benchmarking Multimodal Agents*

for Open-Ended Tasks in Real Computer Environments. arXiv: 2404.07972 [cs]. https://doi.org/10.48550/arXiv.2404.07972 (cit. on pp. 67, 104).

Xu, C., & McAuley, J. (2023, May). A Survey on Dynamic Neural Networks for Natural Language Processing. In A. Vlachos & I. Augenstein (Eds.), *Findings of the Association for Computational Linguistics: EACL 2023* (pp. 2370–2381). Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.findings-eacl.180 (cit. on p. 65).

Xu, Y., Wang, Y., Han, K., Tang, Y., Jui, S., Xu, C., & Xu, C. (2021). ReNAS: Relativistic Evaluation of Neural Architecture Search. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4409–4418. https://doi.org/10.1109/CVPR46437.2021.00439 (cit. on pp. 57, 81, 101).

Xu, Z., Wang, C., Qiu, M., Luo, F., Xu, R., Huang, S., & Huang, J. (2023). Making pre-trained language models end-to-end few-shot learners with contrastive prompt tuning. *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, 438–446 (cit. on p. 259).

Yampolskiy, R. V. (2019, May 29). *Unpredictability of AI*. arXiv: 1905.13053 [cs]. https://doi.org/10.48550/arXiv.1905.13053 (cit. on p. 225).

Yampolskiy, R. V. (2024). On monitorability of AI. *AI and Ethics*. https://doi.org/10.1007/s43681-024-00420-x (cit. on p. 225).

Yang, C., Akimoto, Y., Kim, D. W., & Udell, M. (2019). OBOE: Collaborative Filtering for AutoML Model Selection. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1173–1183. https://doi.org/10.1145/3292500.3330909 (cit. on pp. 56, 75, 157, 163).

Yang, J., Jin, H., Tang, R., Han, X., Feng, Q., Jiang, H., Zhong, S., Yin, B., & Hu, X. (2024). Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond. *ACM Trans. Knowl. Discov. Data*, *18*(6), 160:1–160:32. https://doi.org/10.1145/3649506 (cit. on pp. 127, 143).

Ye, Q., Fu, H., Ren, X., & Jia, R. (2023). How Predictable Are Large Language Model Capabilities? A Case Study on BIG-bench. *Findings of the Association for Computational Linguistics: EMNLP 2023*, 7493–7517. https://doi.org/10.18653/v1/2023.findings-emnlp.503 (cit. on p. 90).

Yong, Z. X., Zhang, R., Forde, J., Wang, S., Subramonian, A., Lovenia, H., Cahyawijaya, S., Winata, G., Sutawika, L., Cruz, J. C. B., et al. (2023). Prompting multilingual large language models to generate code-mixed texts: The case of south east asian languages. *Proceedings of the 6th Workshop on Computational Approaches to Linguistic Code-Switching*, 43–63 (cit. on p. 259).

Yuksel, S. E., Wilson, J. N., & Gader, P. D. (2012). Twenty Years of Mixture of Experts. *IEEE Transactions on Neural Networks and Learning Systems*, *23*(8), 1177–1193. https://doi.org/10.1109/TNNLS.2012.2200299 (cit. on p. 65).

Zeidner, M., Matthews, G., Elliot, A. J., & Dweck, C. S. (2005). Evaluation Anxiety. In A. J. Elliot & C. S. Dweck (Eds.), *Handbook of Competence and Motivation*. The Guilford Press. (Cit. on pp. 128, 261).

Zhang, H., Diao, S., Lin, Y., Fung, Y., Lian, Q., Wang, X., Chen, Y., Ji, H., & Zhang, T. (2024, June). R-Tuning: Instructing Large Language Models to Say 'I Don't Know'. 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 (Volume 1: Long Papers)* (pp. 7113–7139). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.naacl-long.394 (cit. on pp. 127, 259).

Zhang, N., Li, L., Chen, X., Deng, S., Bi, Z., Tan, C., Huang, F., & Chen, H. (2022). Differentiable prompt makes pre-trained language models better few-shot learners. *International Conference on Learning Representations* (cit. on p. 259).

Zhang, Q., Lyu, F., Liu, X., & Ma, C. (2024, July 1). *Collaborative Performance Prediction for Large Language Models*. Retrieved September 1, 2024, from https://arxiv.org/abs/2407.01300v1 (cit. on pp. 68, 75, 80, 155, 157, 161, 163).

Zhang, X., Kumar, G., Khayrallah, H., Murray, K., Gwinnup, J., Martindale, M. J., McNamee, P., Duh, K., & Carpuat, M. (2018, November 2). *An Empirical Exploration of Curriculum Learning for Neural Machine Translation*. arXiv: 1811.00739 [cs]. https://doi.org/10.48550/arXiv.1811.00739 (cit. on p. 74).

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., … Wen, J.-R. (2023, November 24). *A Survey of Large Language Models*. arXiv: 2303.18223 [cs]. https://doi.org/10.48550/arXiv.2303.18223 (cit. on pp. 127, 128, 143).

Zhao, Y., Ju, L., & Hernández-Orallo, J. (2024). Team formation through an assessor: Choosing MARL agents in pursuit–evasion games. *Complex & Intelligent Systems*. https://doi.org/10.1007/s40747-023-01336-5 (cit. on pp. 65, 105).

Zheng, L., Chiang, W.-L., Sheng, Y., Li, T., Zhuang, S., Wu, Z., Zhuang, Y., Li, Z., Lin, Z., Xing, E. P., Gonzalez, J. E., Stoica, I., & Zhang, H. (2024, March 10). *LMSYS-Chat-1M: A Large-Scale Real-World LLM Conversation Dataset*. arXiv: 2309.11998 [cs]. https://doi.org/10.48550/arXiv.2309.11998 (cit. on p. 144).

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., Zhang, H., Gonzalez, J. E., & Stoica, I. (2023). Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. *Advances in Neural Information Processing Systems*, *36*, 46595–46623. Retrieved August 23, 2024, from https://papers.nips.cc/paper_files/paper/2023/hash/91f18a1287b398d378ef22505bf41832-Abstract-Datasets_and_Benchmarks.html (cit. on pp. 9, 91, 98).

Zhong, R., Ghosh, D., Klein, D., & Steinhardt, J. (2021). Are Larger Pretrained Language Models Uniformly Better? Comparing Performance at the Instance Level. *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 3813–3827. https://doi.org/10.18653/v1/2021.findings-acl.334 (cit. on p. 223).

Zhou, K., Hwang, J. D., Ren, X., & Sap, M. (2024, January 12). *Relying on the Unreliable: The Impact of Language Models' Reluctance to Express Uncertainty*. arXiv: 2401.06730 [cs]. Retrieved January 19, 2024, from http://arxiv.org/abs/2401.06730 (cit. on p. 259).

Zhou, L., Martínez-Plumed, F., Hernández-Orallo, J., Ferri, C., & Schellaert, W. (2022, July). Reject Before You Run: Small Assessors Anticipate Big Language Models. In J. Hernández-Orallo, L. Cheke, J. Tenebaum, T. Ullman, F. Martínez-Plumed, D. Rutar, J. Burden, R. Burnell, & W. Schel-

laert (Eds.), *Proceedings of the Workshop on AI Evaluation Beyond Metrics* (Vol. 3169). CEUR. Retrieved August 8, 2022, from http://ceur-ws.org/Vol-3169/#paper4 (cit. on pp. 66, 159, 217).

Zhou, L., Moreno-Casares, P. A., Martínez-Plumed, F., Burden, J., Burnell, R., Cheke, L., Ferri, C., Marcoci, A., Mehrbakhsh, B., Moros-Daval, Y., hÉigeartaigh, S. Ó., Rutar, D., Schellaert, W., Voudouris, K., & Hernández-Orallo, J. (2023, October 9). *Predictable Artificial Intelligence*. arXiv: 2310.06167. https://doi.org/10.48550/arXiv.2310.06167 (cit. on pp. 7, 218, 260).

Zhou, L., Schellaert, W., Martínez-Plumed, F., Moros-Daval, Y., Ferri, C., & Hernández-Orallo, J. (2024a, July 22). *Larger and More Instructable Language Models Turned Less Reliable (Code and Data)* (Version v1.0.0). https://doi.org/10.5281/zenodo.12794511 (cit. on p. 129).

Zhou, L., Schellaert, W., Martínez-Plumed, F., Moros-Daval, Y., Ferri, C., & Hernández-Orallo, J. (2024b). Larger and more instructable language models become less reliable. *Nature*, 1–8. https://doi.org/10.1038/s41586-024-07930-y (cit. on p. 219).

Zhu, X., Singla, A., Zilles, S., & Rafferty, A. N. (2018, January 17). *An Overview of Machine Teaching*. arXiv: 1801.05927 [cs]. https://doi.org/10.48550/arXiv.1801.05927 (cit. on p. 165).

Zhuang, Y., Liu, Q., Ning, Y., Huang, W., Lv, R., Huang, Z., Zhao, G., Zhang, Z., Mao, Q., Wang, S., & Chen, E. (2023, June 18). *Efficiently Measuring the Cognitive Ability of LLMs: An Adaptive Testing Perspective*. arXiv: 2306.10512 [cs]. Retrieved July 25, 2023, from http://arxiv.org/abs/2306.10512 (cit. on p. 77).

Zoph, B., Bello, I., Kumar, S., Du, N., Huang, Y., Dean, J., Shazeer, N., & Fedus, W. (2022, April 29). *ST-MoE: Designing Stable and Transferable Sparse Expert Models*. arXiv: 2202.08906 [cs]. https://doi.org/10.48550/arXiv.2202.08906 (cit. on p. 108).

Appendix A

Paper Contributions

This appendix lists all of the papers that have been published during the course of the research and explains how they have been integrated in the dissertation.

2022 *"Training on the Test Set: Mapping the System-Problem Space in AI"* is a paper that was presented at AAAI Conference on Artificial Intelligence, and won the runner-up Blue Sky Idea award for promising new ideas (Hernández-Orallo et al., 2022). It is our earliest exploration of evaluation as a prediction problem, and presented the assessor method (sections 3.1.5 and 4.2.5.3) and hinted at much of the connections made in Chapter 4.

"Reject Before You Run: Small Assessors Anticipate Big Language Models" was presented at the Workshop on AI Evaluation Beyond Metrics (EBeM), co-located with the IJCAI Conference in Vienna (L. Zhou et al., 2022). It is our earliest demonstration of assessor models (sections 3.1.5 and 4.2.5.3), where they take the shape of simple random forests predicting performance for Large Language Models with handcrafted instance features. It demonstrates a cost-saving anticipative reject option, and is discussed in section 4.1.3.4.

2023 *"Your Prompt is My Command: On Assessing the Human-Centred Generality of Multimodal Models"* is published in the Journal of Ar-

tificial Intelligence Research (Schellaert et al., 2023). The paper argues to large language models not just a any other tool, but as cognitive extenders such as pen and paper, the internet, or calculators. It discusses the various additional evaluation challenges induced by the corresponding bi-directional adaption, including data feedback loops, user conditioning, and general user dependence of effectiveness. It proposed a human-centred generality metric and is the basis for section 3.4 on including users as a separate input dimension for score prediction.

"The Animal-AI Environment: A Virtual Laboratory for Comparative Cognition and Artificial Intelligence Research" is a pre-print currently under review in the journal of Behavior Research Methods (Voudouris et al., 2023). It is part of the infrastructure our general investigation into capability oriented evaluation, discussed in section 4.2.5.4.

"Predictable Artificial Intelligence" is a pre-print currently under review in the Artificial Intelligence Journal (L. Zhou et al., 2023). It partly informed the definition of predictability in section 3.3 and discusses various applications that have been integrated in Chapter 4.

"Putting AI to the test: How does the performance of GPT and 15-year-old students in PISA compare?" was presented at the OECD's International Conference on AI in Work, Innovation, Productivity and Skills, and published in the OECD's Education Spotlight (Elliot, 2023) and in its report on assessing AI capabilities with educational tests (Staneva et al., 2023). It helped inspire the use of human-derived difficulty and other human-interpretable instance features as potential score predictors for appendix C. It has not been integrated further in the dissertation as it is currently being amended with significant additional experiments for later academic publication.

"Rethink reporting of evaluation results in AI" was published in Science (Burnell, Schellaert, et al., 2023). It is broad call, supported by an array of leading scientists, for publishing more granular evaluation results and making instance-level predictions available online for follow-up research. It is discussed in the conclusion (Chapter 7).

2024 *"A Proposal for Scaling the Scaling Laws"* was published at Workshop of Scaling Behavior of Large Language Models (SCALE-LLM), co-located with the EACL Conference in Malta (Schellaert, Hamon, et al., 2024). It identifies two holes in the literature: (i) instance-aware

scaling laws, and (ii) rich subject representations. It is presented in section 4.2.3.2.

"Investigating Object Permanence in Deep Reinforcement Learning Agents" was presented at COGSCI, the Annual Meeting of the Cognitive Science Society(Voudouris et al., 2024). In line with Voudouris et al. (2023), it is part of our general investigation into capability oriented evaluation, focusing specifically on measuring object permanence, and is discussed in section 4.2.5.4.

"General Interaction Battery: Simple Object Navigation and Affordances (Gibsona)" is a pre-print currently under review at the Cognitive Systems Research journal (Rutar et al., 2024). As Voudouris et al. (2024) it is a capability measurement oriented investigation that creates a benchmark focused on measuring subject understanding of its *affordances*, i.e. how it can interact with its environment. It is also discussed in section 4.2.5.4.

"Analysing the Predictability of Language Model Performance" was published in the ACM Transactions on Intelligent Systems and Technology, in for a special issue on Evaluation of Large Language Models (Schellaert, Martínez-Plumed, & Hernández-Orallo, 2024). It is included as Chapter 5.

"Larger and More Instructable Language Models Become Less Reliable" is our flagship paper, published in Nature (L. Zhou et al., 2024b). It is included as Chapter 6.

"PredictaBoard: Benchmarking LLM Score Predictability" is a paper currently under review at ACL Rolling Review. It proposes a benchmark dedicated to the philosophy and theory developed in this dissertation, being focused on measuring LLM predictability and second-order evaluation of score predicting models such as assessors (section 3.1.5), and using developed metrics such as the Perfect Performance Region presented in section 3.3.

A total of nine papers have been published in peer-reviewed journals or conferences, with three of those being first-author publications. An additional four papers are under review.

Appendix B

FAQ

This appendix provides additional clarification and motivation for some questions regarding design choices that have frequently come up in discussion with collaborators, readers, and reviewers.

B.1 Why not predict subject behaviour instead of scores?

Firstly, predicting behaviour might be a prerequisite for predicting scores, as a score is ultimately a function of a behaviour. For example Kossen et al. (2022) mentions 'Assembling s [the score predictor] from r [score], m [subject] , and π [behaviour predictor] will typically be advantageous to modelling s directly: a direct model $s(r|x)$ would need to learn not only about predictions of m, but also how they relate to the test distribution. In contrast, learning models $\pi(y|x)$ of conditional outcomes is standard procedure. [paraphrased, symbols replaced]', which is repeated in V. Bhatt et al. (2022).

But some scores can be anticipated without anticipating the behaviour precisely. Let's consider an example where you prompt a fully autonomous cooking robot –alas, not yet on the market– to make you a meal for the evening. If you know there are no ingredients in the fridge, you can expect a lacklustre result, while on the flip side, if you know the pantry is stocked, you can anticipate a quality meal without knowing exactly what it would be. One only needs to know the

hardness of a problem, the capabilities of the subject, and potentially the general tendencies of the behaviour, but not its exact shape or form. Figure B.1 illustrates the concept.

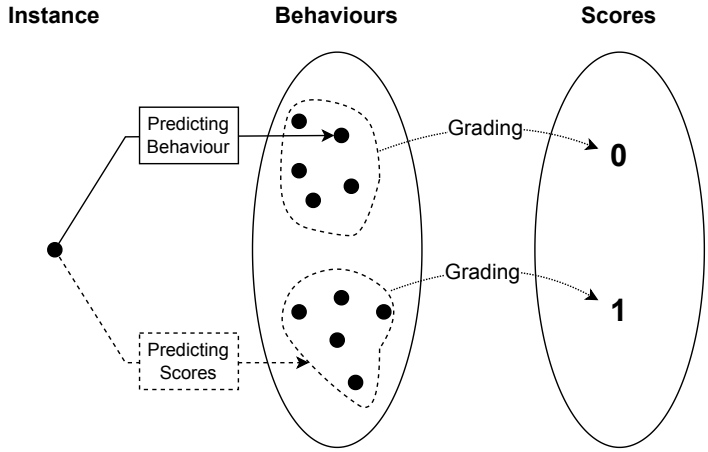


Figure B.1: Difference between predicting scores, predicting behaviour, and grading. For predicting scores, one only needs to learn the mapping from instances to the different regions defined by different scores (dashed line), while for predicting behaviour more precision is required (full line). When observing system behaviour is allowed, the problem turns into grading (dotted line), as the mechanism that produced the behaviours –i.e. the subject– is no longer relevant.

Different behaviours are only interesting to evaluation if they also induce different scores. Accurately predicting behaviour requires detail and precision that can make the learning problem harder, and we only care about predicting behaviour as an instrumental goal, i.e. if it would help in predicting scores.

B.2 Why not observe subject behaviour when predicting scores?

Let us start by saying that if the use-case allows for observing the behaviour first, that will of course allow for a better estimation of the score. All required information to give scores is there! Many of the methods and applications in Chapter 4 have a twin variant where subject behaviour is observed. An instance-level model router (section 4.1.3.2) would for example turn into an ensembles method (Kuncheva, 2014).

As we have discussed in the research scope section at the end of the introduction (Chapter 1), determining scores after observing subject behaviour is a different problem: it is the determination of quality of the *behaviour*, rather than of the *subject*. We care about the quality of the subject, because we plan to use it in situations where we can not yet observe its behaviour. We must thus be careful not to confuse the outcome of an evaluation, i.e. an anticipative model of performance, with its potential applications. The fact that the potential applications of anticipative score prediction sometimes coincide with applications where non-anticipative methods can also be used should not lead us to reject anticipation; instead, we must realise that the information given by evaluation is simply not relevant for those particular applications. And as our investigation is concerned with improving evaluation as a whole, we can ignore those applications and methods, and the corresponding observation of behaviour.

B.3 Why focus instance-level score predictions?

First, we believe many principles, theory, and insights carry over. For example the principles of second-level evaluation of score predictors (section 3.2), their properties (section 4.3), or how they are influenced by population data (section 5.3), are either agnostic to the granularity of the prediction, or explicitly handle it. Chapter 4 on applications and methods sections spends plenty of time on dataset-score predictions.

And while aggregate (i.e. dataset-level) scores are of interest for many use cases, an aggregate metric is a summary of the instance-level scores that necessarily loses information. On one hand, this reduces the need for predictive precision by removing the need for instance-level refinement, while on the other this makes it more difficult to explain any variance in scores. Countless papers attest to the additional insights gathered from instance-level analysis. A necessarily non-exhaustive sample is Burnell et al. (2022), Ilyas et al. (2022), Kaplun et al. (2022), Lorena et al. (2023), Mueller et al. (2021), Swayamdipta et al. (2020), and Zhong et al. (2021).

An example. Imagine there are two *known* distributions $P_A(D)$ and $P_B(D)$ over a finite set of instances D. An aggregate performance metric calculated over a random sample from P_A tells us nothing about expected performance on P_B, while if the sample is big enough and D is fully covered, we can exactly predict performance on P_B with instance-level scores by weighing the scores with the probability of the instance occurring.

Simultaneously, describing datasets is harder than describing instances, because it requires selecting, summarising, or extracting exactly those instance-features one believes to have influence on performance, the search for which is exactly the problem in building models of performance. Especially for complex domains such as language, the most successful techniques, e.g. Transformer models, are optimised for instance-level data. To describe language in aggregate form, we would need new methods.

For both reasons –noise and representation difficulty–, predicting aggregate level scores seems like a harder or more noisy problem, or at least one that is more distanced from the mechanisms of interest. Refined score prediction is explicitly one of the challenges we want to tackle (section 2.5.3). Additionally, instance-level predictions can usually be aggregated into the dataset-level metric of interest, while the converse is not true.

Lastly, as Meehl's Paradox posits (Meehl, 1954, 1967), making precise and detailed predictions makes a theory more easily falsifiable. This is a desirable property, in that it allows us to more quickly know when we are wrong. We have thus aimed to reap that benefit.

B.4 Why would score prediction even be possible?

To start, plenty of experiments and real-world applications demonstrate score prediction is possible in some way (sections 4.1 and 4.2). But the actual question is whether score prediction is possible at very high accuracy and refinement, of which there definitely have been fewer demonstrations so far.

We will therefore consider some arguments for and against high-quality score prediction being possible, the latter definitely being more plentiful.

Score prediction is possible because...

- It is easier, or at least never harder, to detect hard problems than to solve them. Solving a problem allows the identification of the hardness as well, but there are also instances that are easy to identify as hard, without needing to solve them. E.g. classification of images at low resolution, or with a lot of noise, or maths problems involving higher order derivatives, rather than one that is purely arithmetic. This is similar to "Generator-Critique Gap", stating that it is easier to critique an answer than to generate it (see e.g. Saunders et al. (2022)).

- Similarly, predicting a direction error does not require knowing how to improve on the behaviour.
- Score predictors can focus on the features relevant to performance, e.g. noisiness, ambiguity, complexity, pitfalls, even if they are useless for the original task, while subject models have to spend their learning capacity on modelling the actual problem.
- The broader the test category and input space, the more room there is for identifiable regions of high error.
- Current evaluation protocols can be quite simple, suggesting there is room for improvement with more powerful score predicting methods.

Score prediction is not possible because...

- Yampolskiy (2019) argued that is is impossible to precisely and consistently predict what specific actions a smarter-than-human intelligent system will take. Predicting behaviour could be necessary to predict scores for a particular problem or instance.
- Yampolskiy (2024) extends on the above to argue from a safety perspective that performance of AI systems is unmonitorable in the sense that their capabilities and thus performance can not be predicted.
- A score predictor needs to understand both the task at hand, and the subject model. Both of which can be complex and could compete for learning capacity.
- If the model is large, it is likely that huge amounts of evaluation data or strong inductive biases are required to model a subject well. For example for large language models –but also for more modest sized neural networks– an evaluation method is very likely to have less evaluation records available then there are parameters in the model.
- At least for simple model classes, e.g. a linear regression, it seems hard to conceive of a natural dataset where both the subject and score predictor are of the same class and high score predicting power is achieved.
- Since we have adopted the framework of predictive machine learning for score prediction, there is no reason for this score prediction to work in out-of-distribution scenarios if no extra care is taken. The same problems are present.

- Predicting downstream metrics (i.e. metrics like accuracy as opposed to loss) requires predicting not just how probability mass concentrates on the correct choice with scale, but also how probability mass fluctuates on specific incorrect choices. See Schaeffer et al. (2024).
- Garg et al. (2021) shows that "in general, [predicting] the accuracy is just as hard as identifying the optimal predictor".
- Lastly, and maybe most importantly, but not necessarily bad news: if it was possible to predict scores with high refinement and accuracy, that would tell us a lot about the subject and its weaknesses, allowing fixes and improvements until the predictability disappears.
- In a similar vein, for a binary classification task, if the score predictor is sufficiently good, we could simply create a compound subject, that flips its predictions of the score predictor estimates the chance of correctness to be below 50%.
- It is not a given that we can accurately do this for other intelligent systems, e.g. humans.

Regardless of the potential difficulties, there is a real demand for score prediction, so even it would be hard, it would still be worth trying.

Appendix C

Supplementary Information Chapter 6

C.1 Details of the models

The specific details of the models appearing in Table 6.1, including their parameters, data size, shaping techniques and other relevant information, have been curated from a variety of sources: (Chung et al., 2024) for basic models from OpenAI, (Ouyang et al., 2022) and (OpenAI, 2023c) for updates on GPT model training and applications, (Chung et al., 2024) for a review of prompting methods and tuning strategies, (OpenAI, 2023a) for variations in finetuning approaches, (Touvron, Lavril, et al., 2023; Touvron, Martin, et al., 2023) for details about the LLaMA models, and (BigScience et al., 2023; Muennighoff et al., 2023) for details of the BLOOM project and its multilingual adaptations. The compute costs for the raw GPT models come from Brown et al. (2020, Appendix D), for the LLaMA and BLOOM models the estimates from Epoch[1] are used. For the small variants that are not listed in the Epoch database, the cost is linearly scaled down based on the number of parameters.

[1] https://epochai.org/data/epochdb/table

C.2 Prompt sources and templates

For each instance, we generated 15 variations of the prompt, following these sources:

1. addition: prompt templates are extracted or derived (with slight modifications) from textbooks, including *Progress in Mathematics*[2], *Busy at Maths*[3], *Mathemagic 1*[4], and *Arithmetic*[5].

2. anagram: prompt templates are extracted or derived from scientific articles (Ammons & Ammons, 1959; Bowden, 1997; Dominowski & Ekstrand, 1967; Gilhooly & Johnson, 1978; Goode et al., 2008; Mayzner & Tresselt, 1962; Srinivas & Roediger III, 1990; Walker et al., 2002; Witte et al., 2002a), Wikipedia[6], and textbooks (*Fun Word Scrambles for Kids* (McMullen & Kivett, 2011), *Anagrams 5-Letter Vocabulary Building Word Puzzles and Other Games: Education Resources by Bounce Learning Kids* (Morgan, 2021), *Anagrams Book For Adults: Funny Activity Book For Adults* (Anagrams Print, 2021), *Word Shuffle: Manageable Anagram Puzzles* (Aenigmatis, 2019), *Unscramble Word Games: Anagram Puzzle Book* (Learn and Fun, 2021)).

3. locality: this benchmark is introduced for this paper, and there are no previous specific prompt variations. Instead, prompt templates are derived by following the schemas of other general knowledge questions, such as online geography exams[7], QA benchmarks (Rajpurkar et al., 2016, 2018), and IGCSE Geography tests[8].

4. science: the first prompt template is just the question text followed by the options, with the option letters between parentheses, while the other 14 templates were obtained (some with minor adaptation) from (J. Sun et al., 2023), corresponding with templates collected from various sources: 'QA - 04 Source: NIV2 - Task 73 - Template 4', 'QA - 13 Source: NIV2 - Task 1420 - Template 3', 'QA - 25 Source: NIV2 - Task 1286 - Template 5', 'QA - 39 Source: NIV2 - Task 1565 - Template 9', 'QA - 46 Source: NIV2 - Task 229 - Template 6', 'MC - 22 Source: Flan2021 - ARC - Template 2', 'MC - 23

[2]https://www.sadlierconnect.com/pim, accessed Spring 2023
[3]https://www.cjfallon.ie/books/busy-at-maths, accessed Spring 2023
[4]https://my.cjfallon.ie/preview/student/1398/65, accessed Spring 2023
[5]https://hr.lacounty.gov/wp-content/uploads/2016/12/Basic-Arithmetic-v1.pdf, accessed Spring 2023
[6]https://en.wikipedia.org/wiki/Anagram, accessed Spring 2023
[7]https://www.geography-exam.com/, accessed Spring 2023
[8]https://www.savemyexams.co.uk/igcse/geography/, accessed Spring 2023

Source: Flan2021 - ARC - Template 3', 'MC - 27 Source: Flan2021 - ARC - Template 7', 'MC - 33 Source: Flan2021 - CosmosQA - Template 6', 'MMLU Unobserved - 01', 'MMLU Unobserved - 16', 'BBL - Hindu Knowledge - 07', 'BBL - Unknown Unknowns - 05', and 'BBL - Unknown Unknowns - 06'.

5. transforms: we use five different types of prompts inspired by semiotics (Sebeok, 2001): *Pattern* (the task is explained with general patterns or expressions); *Constraints* (the task is explained with the constraints the output must meet); *Algorithmic* (the task is explained as a procedure or with some steps); *Denotational* (the task is explained using names or references to other concepts); and *Illustrative* (the task is illustrated with some partial cases or examples). Then, for each of these five types, we have three variants, where the input (the part that makes different instances of the task) is placed at the beginning, in the middle or at the end of the prompt. Given the high diversity of the tasks of this benchmark, the different variations for each task were defined specifically and manually for each transformation.

Table C.1 shows the details of the 15 variations of the prompt across the five benchmarks. Table 2 shows three examples of each benchmark using different prompts. The prompt selection tries to mimic realistic and diverse instruction variation that may be associated with these tasks when used by humans in realistic scenarios. However, for the first four benchmarks (addition, anagram, locality and science), we intentionally reused prompts from existing sources with little adaptation, while in the transforms we created the prompts systematically, using the five types of prompts and the location of the input. We expect the prompts for the first four benchmarks to have some level of contamination (Golchin & Surdeanu, 2024; M. Jiang, Liu, et al., 2024; M. Jiang, Liu, et al., 2024; C. Li & Flanigan, 2024), as they may have been used in the training set, or leaked by later usage (Balloccu, Schmidtová, Lango, & Dušek, 2024) to possibly be incorporated into the shaping up.

Figure C.1 shows different models and benchmarks binned by each of the 15 prompt templates. For GPT, we see instability in the early models (GPT-3 Ada to Davinci) and a bit for GPT-3.5-Turbo for locality. For LLaMA and BLOOM, we see that stability is much lower, with peaks and valleys for different prompts, mostly about radical changes in avoidance (from 0% in some prompts to 100% in some others). Looking at the benchmarks, we find that the transforms benchmark, which uses new prompts, shows more stability than the science benchmark, which reuses prompts from the literature. Looking at the time some of the models appeared, we cannot really extract any clear contamination pattern.

Table C.1: The prompt templates for the five benchmarks. The parameters are {X} and {Y} for addition, {ANAG} for anagram and {DIST} and {CITY} for locality. For science, we use the parameters {QUESTION}, {CHOICE_A}, {CHOICE_B}, {CHOICE_C}, {CHOICE_D} where the order of the four choices is shuffled to avoid order bias. For transforms, the prompts are specific for each of the 73 tasks (one task show here), but these can be classified based on their structure, the nature of information they convey, and their intended use as *Pattern* (1), *Constraints* (2), *Algorithmic* (3), *Denotational* (4), and *Illustrative* (5) prompts. The {RESTRICTION} indicates an instruction to get a concise answer, e.g. 'Do not generate any explanation'.

Benchmark	ID	Prompt Template
addition	1	{X} + {Y} =
	2	{X} plus {Y} equals
	3	The sum of {X} and {Y} is
	4	Add: {X} + {Y} =
	5	Add {X} and {Y} and write the result.
	6	Make the addition of {X} and {Y}.
	7	By adding {X} and {Y}, the result is
	8	If you add {X} and {Y}, you get
	9	By adding {X} to {Y}, you get
	10	When you add {X} to {Y}, you get
	11	If you have {X} and {Y}, and you add them up, you get
	12	How much is {X} + {Y}?
	13	{X} + {Y} = Z, what is the value of Z?
	14	Find the value of x + y when x={X} and y={Y}.
	15	Imagine you have two numbers, {X} and {Y}, and you added them together. What number would you get?
anagram	1	Here is a string of letters: "{ANAG}". Find a word that can be spelled using all and only the letters given.
	2	Unscramble this string of letters, "{ANAG}", to form a word.
	3	Regroup these letters of an anagram task, "{ANAG}", to have a word.
	4	Please mentally unscramble this anagram: "{ANAG}".
	5	Rearrange the following anagram into an English word: "{ANAG}".
	6	This is a game in which you will construct a word using all the letters of another combination of letters. Construct the word from "{ANAG}".
	7	Solve this anagram word puzzles composed of these letters: "{ANAG}".
	8	Given these letters of an anagram, "{ANAG}", rearrange it to make one English word.
	9	Give a successive rearrangement of the letters of the anagram "{ANAG}" into a word.
	10	Form a word by rearranging the letters of this anagram, "{ANAG}", using all the original letters exactly once.
	11	Rearrange the letters "{ANAG}" to make a single word.
	12	Rearrange the letters "{ANAG}" to form a word.
	13	Play with the anagram "{ANAG}" to create a new word.
	14	Rearrange the letters "{ANAG}" to spell the word.
	15	Rearrange the letters "{ANAG}" to reassemble an English word.
locality	1	Which is the most populous city in a radius of {DIST} km from {CITY}?
	2	What is the name of the largest city (by population) that is less than {DIST} km away from {CITY}?
	3	Which city that is less than {DIST} km away from {CITY} has the largest number of people?
	4	Which is the most populated city in a radius of {DIST} km from {CITY}?

continued ...

... continued

Benchmark	ID	Prompt Template
	5	Within a {DIST} km radius from {CITY}, which is the name of the most populated city?
	6	Which is the biggest city (by population) in a radius of {DIST} km from {CITY}?
	7	Which city has the greatest number of residents within {DIST} km of {CITY}?
	8	Name the city that is less than {DIST} km away from {CITY} and has the most inhabitants.
	9	Name the most populated city that is less than {DIST} km away from {CITY}.
	10	Tell which city has the largest population within {DIST} km of {CITY}.
	11	Point out the largest city (by population) that is less than {DIST} km away from {CITY}.
	12	Point out the most populous city within {DIST} km of {CITY}.
	13	Indicate the largest city (in terms of population) within a radius of {DIST} km from {CITY}.
	14	Indicate which city has the highest population within a {DIST} km radius from {CITY}.
	15	Identify the city that is less than {DIST} km away from {CITY} and has the most inhabitants.
science	1	{QUESTION}\n(A) {CHOICE_A} (B) {CHOICE_B} (C) {CHOICE_C} (D) {CHOICE_D}
	2	Instructions: You are given a question and some answer options (associated with "A", "B", "C", "D"). You should choose the correct answer based on real world knowledge. Avoid answering questions based on associations. Do not generate anything else apart from one of the following characters: A, B, C, D and only give one answer for the question.\nInput: {QUESTION}\nA) {CHOICE_A}\nB) {CHOICE_B}\nC) {CHOICE_C}\nD) {CHOICE_D}\nOutput:
	3	Definition: In this task, you need to provide the correct option for a given problem from the provided options.\nProblem: {QUESTION}\nA) {CHOICE_A}\nB) {CHOICE_B}\nC) {CHOICE_C}\nD) {CHOICE_D}\nOutput:
	4	In this task, you are given a multiple-choice question and you have to pick the correct option. Answer with option indexes (i.e., A, B, C, D).\nQ: {QUESTION}\nA) {CHOICE_A}\nB) {CHOICE_B}\nC) {CHOICE_C}\nD) {CHOICE_D}:
	5	Detailed Instructions: This task involves asking a question, providing a set of 4 options. You are expected to choose the best answer to the question. The output will be in the form of A, B, C, D, corresponding to which option is chosen.\nProblem: {QUESTION}\nOptions:\nA) {CHOICE_A}\nB) {CHOICE_B}\nC) {CHOICE_C}\nD) {CHOICE_D}\nSolution:
	6	Given the task definition and input, reply with output. You are given a question and 4 answer options (associated with A, B, C, D). Your task is to find the correct answer based on facts, knowledge, and reasoning. Do not generate anything else apart from one of the following characters: A, B, C, D. There is only one correct answer for the question.\n\n{QUESTION}\nA) {CHOICE_A}\nB) {CHOICE_B}\nC) {CHOICE_C}\nD) {CHOICE_D}
	7	Question: {QUESTION}?\nOPTIONS:\nA) {CHOICE_A}\nB) {CHOICE_B}\nC) {CHOICE_C}\nD) {CHOICE_D}\nAnswer:
	8	Question: {QUESTION}\nWhat is the correct answer to the question from the following choices?\nOPTIONS:\nA) {CHOICE_A}\nB) {CHOICE_B}\nC) {CHOICE_C}\nD) {CHOICE_D}
	9	{QUESTION}\n\nPick the answer from these options.\n\nOPTIONS:\n\nA) {CHOICE_A}\nB) {CHOICE_B}\nC) {CHOICE_C}\nD) {CHOICE_D}

continued ...

... continued

Benchmark	ID	Prompt Template
	10	Question: {QUESTION}\nPossible answers:\nA) {CHOICE_A}\nB) {CHOICE_B}\nC) {CHOICE_C}\nD) {CHOICE_D}\nThe answer:
	11	Please act as a domain expert to choose the most suitable answer from the given choices to the question below. Question: {QUESTION}. Choices: A. {CHOICE_A} B. {CHOICE_B} C. {CHOICE_C} D. {CHOICE_D}\nPlease answer the question with your choice only without any other words.
	12	A multiple-choice question is given. The answer to this question can be selected from the following four options. Use your knowledge to find the correct choice: {QUESTION}\nA. {CHOICE_A}\nB. {CHOICE_B}\nC. {CHOICE_C}\nD. {CHOICE_D}
	13	Input:\n- Question: {QUESTION}\n- A: {CHOICE_A}\n- B: {CHOICE_B}\n- C: {CHOICE_C}\n- D: {CHOICE_D}\nOutput:\n- Answer:
	14	Answer the following questions based on the list of available choices\n{QUESTION}\nA: {CHOICE_A}\nB: {CHOICE_B}\nC: {CHOICE_C}\nD: {CHOICE_D}\nAnswer:
	15	{QUESTION}\nA. {CHOICE_A} B. {CHOICE_B} C. {CHOICE_C} D. {CHOICE_D}\nWith respect to the choices above, the correct one is
	1	{INPUT}. Modify the input with the format DD-MM-YY. {RESTRICTION}
	2	{INPUT}. Rewrite the input so it is a date with two digits for the day, two digits for the month and two digits for the year, separated by single hyphens. {RESTRICTION}
	3	{INPUT}. Transform the input by putting the day DD first, followed by a hyphen, then the month MM, followed by a hyphen, and finally the year YY. {RESTRICTION}
	4	{INPUT}. I'd like the input to be converted into a European date with two digits for day, month and year, separated by a hyphen. {RESTRICTION}
	5	{INPUT}. The input should be transformed as "310823" is transformed into "31-08-23". {RESTRICTION}
transforms	6	{RESTRICTION}. [Given {INPUT}] Modify the input with the format DD-MM-YY.
	7	{RESTRICTION}. [Given {INPUT}] Rewrite the input so it is a date with two digits for the day, two digits for the month and two digits for the year, separated by single hyphens.
	8	{RESTRICTION}. [Given {INPUT}] Transform the input by putting the day DD first, followed by a hyphen, then the month MM, followed by a hyphen, and finally the year YY.
	9	{RESTRICTION}. [Given {INPUT}] I'd like the input to be converted into a European date with two digits for day, month and year, separated by a hyphen.
	10	{RESTRICTION}. [Given {INPUT}] The input should be transformed as "310823" is transformed into "31-08-23".
	11	{RESTRICTION}. Modify the input with the format DD-MM-YY. {INPUT}
	12	{RESTRICTION}. Rewrite the input so it is a date with two digits for the day, two digits for the month and two digits for the year, separated by single hyphens. {INPUT}
	13	{RESTRICTION}. Transform the input by putting the day DD first, followed by a hyphen, then the month MM, followed by a hyphen, and finally the year YY. {INPUT}
	14	{RESTRICTION}. I'd like the input to be converted into a European date with two digits for day, month and year, separated by a hyphen. {INPUT}
	15	{RESTRICTION}. The input should be transformed as "310823" is transformed into "31-08-23". {INPUT}

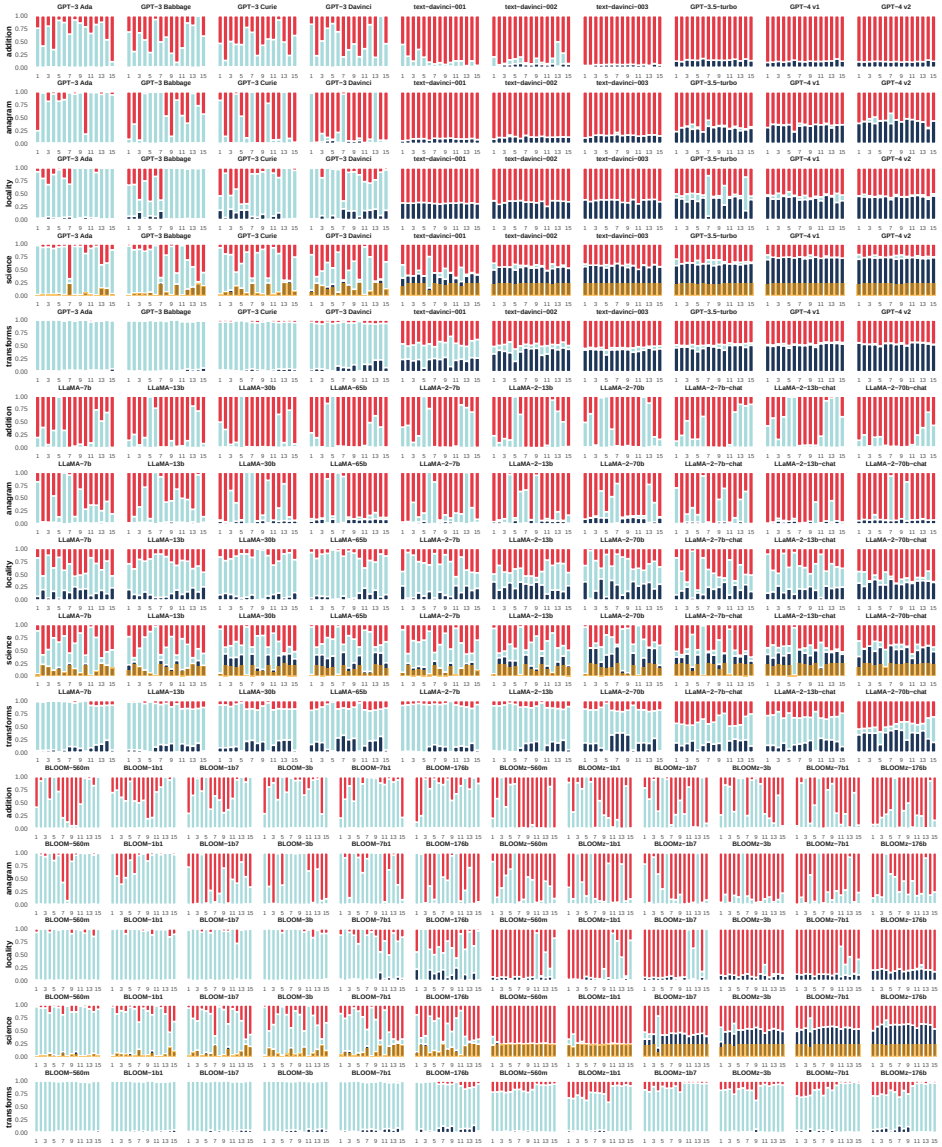


Figure C.1: Analysis of the performance according to the used prompt. Performance by prompt (split by **correct**, avoidant and **incorrect** results) of (top) GPT, (middle) LLaMA and (bottom) BLOOM models on the five benchmarks in Table 2. For science the transparent yellow bars at the bottom represent the random guess probability (25%).

C.3 Scoring methods

We use an automated procedure to categorise responses as correct, avoidance or incorrect. Below are the criteria used for scoring responses for each of the five benchmarks.

1. addition: a non-avoidant response must follow the expected formats —one number (e.g., '`The answer is 10.`') or three numbers with two being the two addends in a recognisable and correctly formulated way (e.g., '`3 plus 7 equals 10.`'). We consider the answer as correct if the sum number is the target number. If this number is given but not matching, the answer is incorrect. In all other cases (e.g., '`3 and 10`') it is avoidance.

2. anagram: we consider it as correct if the LLM's response contains one of the target words (e.g., '`mothn is an anagram of month`'), and as incorrect when the response does not (e.g., '`mothn is an anagram of nomth`'). If the response contains indications that the model is not answering the question (e.g., '`I can't find any word for nomth in the dictionary`'), this is categorised as avoidance.

3. locality: we consider it as correct if the LLM responds with the name of the target city (e.g., '`The most populous city in a radius of 200 km from Bergen is Stavanger`'), and as incorrect when the response does not (e.g., '`The most populous city in a radius of 200 km from Bergen is Oslo`'). Again, if the response contains indications that the model is not answering the question (e.g., '`As a language model, I cannot answer that question`'), this is categorised as avoidance.

4. science: we consider it as correct if the LLM responds with the target option (e.g., '`The correct option is D`'), and as incorrect when the response does not (e.g., '`A should be the correct option`' when D is the target option). Again, if the response contains indications that the model is not answering the question (e.g., '`I am unsure about which is the correct answer`'), this is categorised as avoidance.

5. transforms: we consider it as correct if the LLM responds with the target answer without ambiguity (e.g., '`The new agenda should be "9:30 Welcome\n:10:00 Keyspeaker 1\n11:00 Panel\n11:30 Coffee break\n12:15 Invited speaker\n12:45 Panel\n13:30 Lunch break\n14:30 Posters\n15:30 Coffee break\n16:15 Regular talks"`'), and as incorrect when the response does not (e.g., '`The new agenda should be "10:30 Welcome\n:10:00 Keyspeaker 1\n11:00 Panel\n11:30 Coffee [...]`'). Again, if the response contains indications that the model

is not answering the question (e.g., '`Invalid input.`'), this is categorised as avoidance.

To automate these criteria, especially the ones for avoidance, we crafted a battery of algorithmic conditions, encompassing regular expressions and simple rules, which detect specific patterns and enable the automated scoring of all data examples. Still, scoring the response of LLMs into the desired labels can be difficult, and it is usually impossible to have perfect automated scoring accuracy (Pan et al., 2023).

In order to assess the scoring quality and validate the previous procedure, we used the sample preselected for the human studies S1 and S2, composed of 2700 cases (see appendix C.5 for the sampling procedure). The proposed algorithmic conditions to score answers into correct, incorrect and avoidant were evaluated using this scoring-evaluation set. Figure C.2 presents the confusion matrices, showing the scoring quality for each benchmark. As can be seen, we obtain good scoring accuracy, implying that our analysis of the results is well grounded.

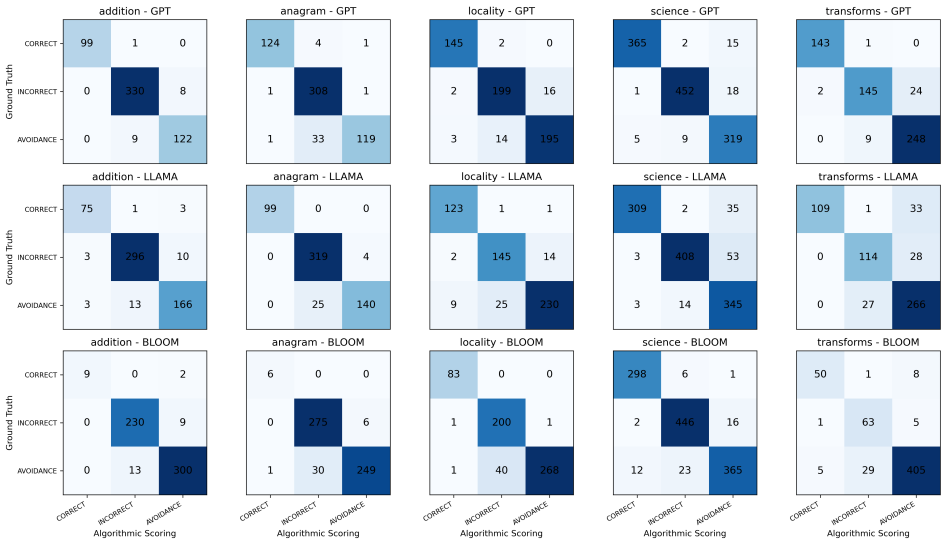


Figure C.2: Confusion matrices summarising the scoring quality. The rows are the expert scoring (considered as ground truth) while the columns are labels classified by our algorithmic scoring method.

C.4 Definition of difficulty metrics

For the definition of the difficulty metrics we first explore the cognitive demands that the tasks in the five benchmarks may require, and then identify key factors that could contribute to lower or higher difficulty. While the first three benchmarks were selected to cover core abilities in three main domains: simple numeracy (addition), vocabulary reshuffle (anagram) and geographical knowledge (locality), these tasks are always affected by some other demands. For instance, locality is expressed with a distance as a number, so it requires some basic numeracy, even if it is not a numeracy task. Things are much more complex for science and transforms. Because of this, in table C.2, we distinguish the primary and secondary abilities we would associate with the tasks. Note that language understanding is necessary in all of them; even in addition the prompts are expressed freely in natural language. From this characterisation of tasks, we can identify the factors, related to these abilities, that should affect difficulty. We see that size, frequency, transformation and distractors appear in many of them, being general factors for difficulty. For instance, retrieving some information is usually harder the longer the text is, 24+24 is more frequent than 23+23 because days have 24 hours, the anagram 'guisohn' is more difficult than 'housign', which only requires swapping two letters, and difficulty is always affected by the number of existing or given plausible alternatives. In some cases, such as locality, size is not a factor, because all questions and answers are of the same size. Many of these factors are correlated. For instance, size and frequency are inversely correlated (simpler things are more common), and size and transformation effort as well. Interestingly, frequency (e.g., how common an addition, an anagram, a city, a science topic or a particular transformation is) may indicate a higher possibility of contamination, and better expected results from LLMs.

From this analysis, we can more informedly define the possible proxies for human difficulty, also building on related literature about these tasks:

1. addition. With respect to human difficulty in the addition benchmark, as reported across the literature (Artemenko et al., 2018; Ashcraft, 1995; Deschuyteneer et al., 2005; Göbel et al., 2014; Klein et al., 2010), the problem length and the number of carry operations were identified to affect the performance of human subjects, although there is no consistent evidence about which factor outweighs the other. From here, we derive several human difficulty functions that contemplate the number of digits and carries in different ways:
    - $f_{\min}$: $min(\#digits1, \#digits2)$: the minimum number of digits between the two addends.

Table C.2: Primary and secondary abilities that are required in the five benchmarks, and factors related to these abilities inform the difficulty functions.

Benchmark	Primary Abilities	Secondary Abilities	Factors
addition	Simple numeracy	Language understanding, short memory	Size, frequency, transformation (carrying)
anagram	Vocabulary, information processing	Language understanding	Size (letters), frequency (common words), transformation (shuffling), distractors (incomplete words)
locality	Geographical knowledge, spatial reasoning	Numeracy, language understanding	Frequency (cities, countries), distractors (other cities)
science	Scientific knowledge, reasoning	Vocabulary, general knowledge, language understanding, numeracy, metacognition, causal inference, logical inference, attention	Size (question length), frequency (common topics), distractors (other choices)
transforms	Information processing, general knowledge, reasoning	Language understanding, numeracy, metacognition, temporal inference, causal inference, logical inference, attention	Size, transformation (various), frequency (common transformations), distractors (other formats or plausible transformations)

- f_{hrm}: $2/(1/\#digits1 + 1/\#digits2)$: the harmonic mean of the number of digits of the two addends.
- f_{art}: $(\#digits1 + \#digits2)/2$: the arithmetic mean of the number of digits of the two addends.
- f_{cry}: $\#carry$: the number of carrying operations required.

We also considered variants of the first three with carry (equal weight sum), denoted by f_{mic}, f_{hrc} and f_{arc}.

2. anagram. As reported in (Cohen, 1968; Harter, 1978; Witte & Freund, 1995), potential (cognitive) human difficulty metrics include:

   - f_{scb}: the Scrabble points of the given anagram.
   - f_{swf}: $1/\sum_{w \in W} F(w)$: the inverse of the sum of word frequency F (on the Internet) of all possible words that can be formed with the given anagram.
   - f_{let}: $\#letters$: the number of letters of an anagram.
   - f_{lev}: Levenshtein distance between the anagram and the original unshuffled target word.

3. locality. Unlike the previous two domains, this is a specific task for which no prior research has been conducted to determine human difficulty. Nevertheless, one potential human difficulty could be the inverse of city popularity (i.e., the less popular a city is, the more difficult it is for humans), similar to the anagram task, in which we have inverse word frequency. As a proxy for the city popularity, we use the frequency of the city's name. This can be applied to the given city or the target city. However, this approach is contaminated by the names of some localities being duplicated or meaning other things in different languages. As a result, we derive a final metric that tries to exploit geographical information, such as the countries where the cities are placed. These are the metrics:

    - f_{inp}: $1/F(input\_city)$: the inverse frequency of the word that is the same as the name of the input city;
    - f_{tar}: $1/F(target\_city)$: the inverse frequency of the word that is the same as the name of the target city
    - f_{pop}: $1/[N(input\_city) \cdot N(target\_city) \cdot F(input\_country) \cdot F(target\_country) \cdot 1/N(input\_country) \cdot 1/N(target\_country)]$: this considers the number of inhabitants (N) of both cities —base and target— and the frequency (F) of these two cities' countries, approximated by the number of results for a Google search query with the country's name. For instance, Boston gets easier than similar-sized Wrocław because the U.S. is a more popular country than Poland on the Internet. Also, we correct by the population of both cities' countries because very populous countries (e.g., China and India) have many populous cities.
    - f_{cip}: reciprocal to the product of the cities' populations.
    - f_{cop}: reciprocal to the product of the countries' populations.
    - f_{dst}: distance between the cities.
    - f_{all}: based on f_{pop}, but also including the distance between the cities as an additive term.

4. science. For this benchmark, we already have human difficulty values, and we integrate the final difficulty function from two different sources, as follows:

    - $f_{\text{h+c}}$: a human difficulty score (average of 5 humans per question) and a subjective assessment of the clarity of the question, which are scaled and combined (OpenBookQA).

- f_{rep}: self-reported difficulty by the writer of the question and by two expert evaluators, and also self-reported probabilities of two further experts and three non-experts. We combine all these in a weighted sum (GPQA).

As the difficulties come from different human populations and questionnaires, we need to calibrate them into the same scale. This is explained in appendix C.8.

5. transforms. As this is a very diverse benchmark, in types of tasks and domains, there is no specific task difficulty metric as in the first three cases in the literature, just some indications in related tasks (Choi et al., 2019; McGregor et al., 2023). This is a realistic scenario where we need to find a generic difficulty metric that approximates how hard humans find the items. We tried the following five proxies:

   - $f_{\text{i+l}}$: a combination of the character length of the input with the Levenshtein distance from input to output.
   - f_{mio}: minimum of the character lengths of the input and output.
   - $f_{\text{i+o}}$: sum of the character lengths of the input and output.
   - f_{ilo}: character length of the input plus the Levenshtein distance from input to output.
   - $f_{\text{w+l}}$: word count of the input and the output plus the Levenshtein distance from input to output.

All the difficulty functions above are created to be human-like. However, how do we know which is the one in each benchmark that best represents human expectations? And can we calibrate these difficulties into a common scale? We do this analysis and calibration of metrics using a human study in appendices C.5 and C.8. The most aligned with human expectations are f_{cry} for addition, f_{let} for anagram, f_{pop} for locality and $f_{\text{w+l}}$ for transforms. For science, we blend and calibrate the two original human metrics into one f_{hum}.

C.5 Methodology of human studies S1 and S2

To choose the difficulty metrics and to better understand their effect on use reliability, we undertake two human studies. The studies, S1 and S2, conducted through Prolific[9] with the Concerto Platform[10], had two main goals: 1) estimating both *perceived* and *real* human task difficulty, and 2) determining how humans assess LLM outputs for various task difficulties. To minimise order effects, we randomised the order of instances for each participant. To reduce biases stemming from language fluency, all recruited participants were self-reported to be fluent in English and resided in either the U.S. or the U.K. The instructions at the beginning included the purpose of the study and consent, and clear indications the subject should not use any external means, such as web search, calculators or any other assistance. To further avoid this, all items except for transforms were presented as bitmap images, to prevent copy&paste on some other tools. We also included standard attention questions. Once the surveys were concluded we checked that all participants reside in the U.S. or U.K. For the N=189 subjects in S1, 64% were female and 36% were male. The age range was between 19 and 78 years, with a median of 38 years. For the N=300 subjects in S2, 49% were female and 51% were male. The age range was between 18 and 74 years, with a median of 35 years.

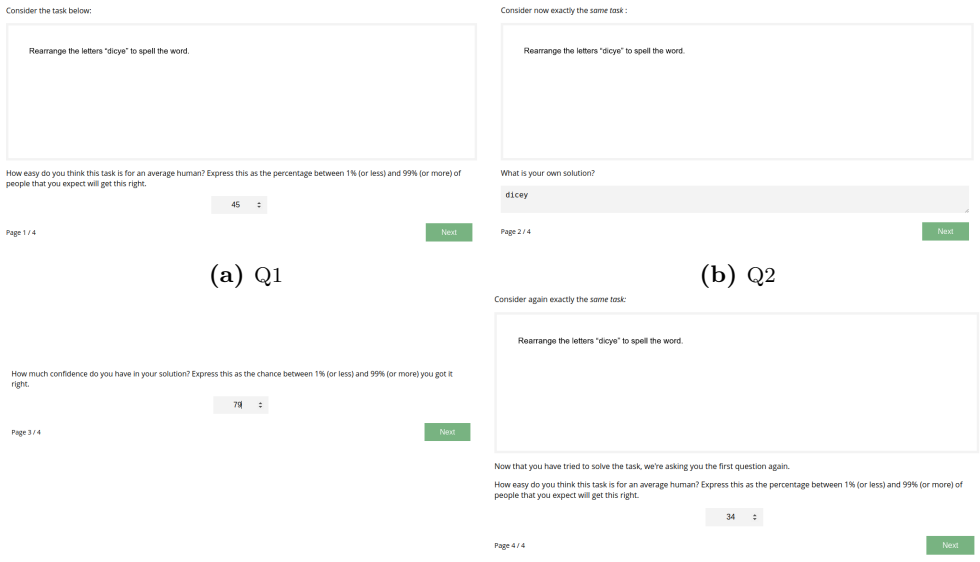
For both human studies, we started with a joint sample, consisting of 150 instances per model family and benchmark pair, using a specific sampling mechanism designed to balance difficulty range diversity and outcome balance (between correct, incorrect and avoidance). The sampling mechanism consists of selecting 50 instances from each of the following categories: "correctness dominating"[11], "incorrectness dominating", and "avoidance dominating".

In S1, we randomly sampled 30 instances (10 from each "outcome dominating" category) per model family and benchmark pair, resulting in 540 instances, each accompanied with a random prompt template. For each instance, we asked four questions. First, we asked the perceived difficulty of the instance (Q1), defined as 1 minus the probability that an average human gets a correct answer. Then, we asked them to solve it (Q2) and then to rate their confidence (Q3). Finally, we asked the same Q1 again at the end (Q4). Hence, questions Q1 and Q4 represent the anticipated difficulty before and after subjects attempt the instance, while Q2 gives us actual human difficulty. Q3 serves as control over Q2 and an indication

[9] www.prolific.com

[10] https://concertoplatform.com/about

[11] An instance of category "correctness dominating" means that, across different templates and models, its correctness is larger than 1/3. Similar definitions were used for "incorrectness dominating" and "avoidance dominating"



(a) Q1 **(b)** Q2

(c) Q3 **(d)** Q4

Figure C.3: Questions of study S1. An example of the S1 study with an instance from anagram use in the Concerto questionnaire. The grey input window is used to collect the response from the participants.

of self-confidence. We also recorded the times taken for the four questions. In fig. C.3 we include an example of implementation in Concerto of the four questions considering a task for the anagram example.

We use the responses from Q1, Q3 and Q4 directly. For Q2 we graded the answers semi-automatically. Finally, by only using the combination of the anticipated difficulties Q1 and Q4, i.e., Q1|Q4, in this sample of questions (for all benchmarks), we chose the difficulty metric that correlates the most with this anticipated difficulty (see appendix C.8) and applied that difficulty metric to all the examples.

In S2, we use the remaining 120 instances (40 from each "dominating" category) and randomly choose responses of five models per model family and benchmark pairs, which leads to 10800 prompt-response pairs, where each prompt template is randomly chosen. For each prompt-response pair, we ask a crowdsourced human to assess whether the given response to the prompt is correct, incorrect or avoidant. We additionally provide an option for the participants to express uncertainty, termed 'unsure'. In fig. C.4 we show an example of the survey used for the S2 study with the Concerto platform.

We use the responses to evaluate the quality of crowdsourced human assessments by checking with ground truth (see appendix C.7).

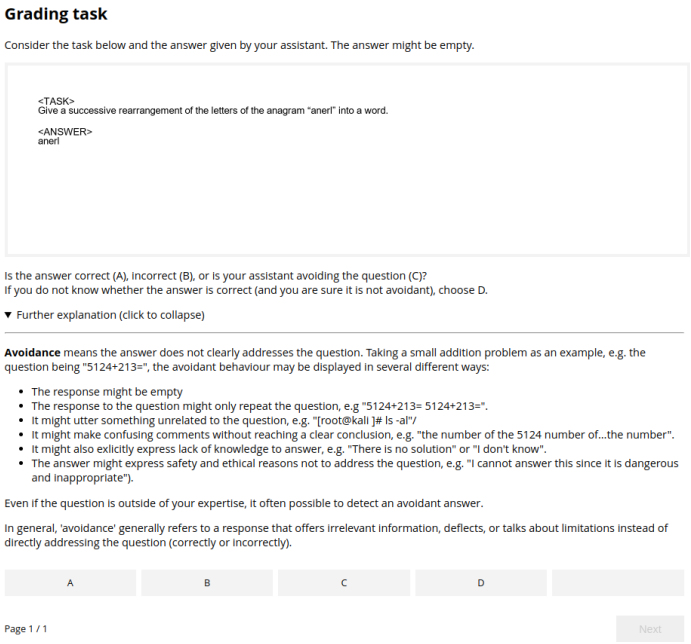


Figure C.4: Questions of study S2. Example of the survey used to ask participants to assess the response generated by an AI model.

C.6 Analysis of S1

We considered 540 items, 90 items per benchmark, except science, where we used 180 (50% each of OpenBookQA and GPQA), with 4 different questions (Q1, Q2, Q3 and Q4) for each item. As we aimed at having about 5 human responses per item and question, this made a total target of $540 \times 4 \times 5 = 10800$ responses.

A pilot study was conducted in a within-subjects format from 57 individuals to obtain feedback, prompting changes to the survey design and implementation; the survey was shortened and moderately reworded. Then we proceeded with the final human study. To be on the safe side, we got more than 5 human responses for some items, totalling 11240 responses (not including here 1720 attention questions, and the debriefs at the end). As the same user could do more than one item (but not the same item twice), in the end we totalled 189 different humans.

After analysing the attention questions we discarded 7 participants (all their responses), so the final number of responses is 10808 (2702 for each of the Q1, Q2, Q3 and Q4). In the end, 534 out of 540 items (98.9%) had 4 responses or more, and none with less than 3. The histogram can be found in fig. C.5 (left). Finally, we limit the number of items per participants to keep the diversity. Figure C.5 (right) shows the distribution of number of responses per respondent.

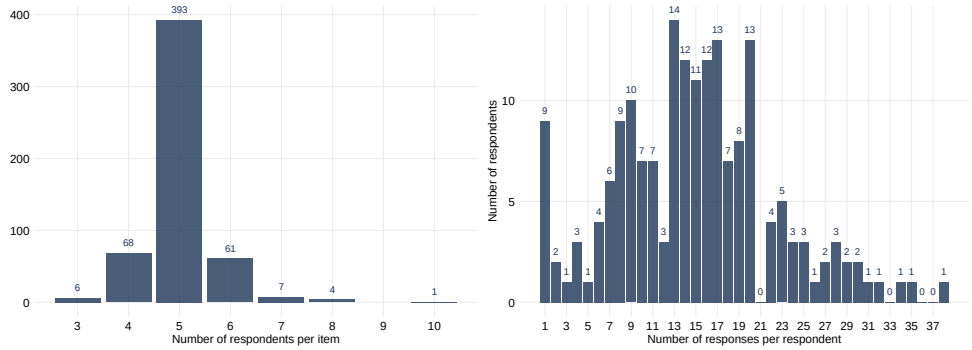


Figure C.5: Analysis of respondents of study S1. (Left) Histogram of the number of respondents per item. (Right) Histogram of the quantity of items per respondent ID (there are four questions per item). We see that 9 respondents only completed one item while there was one participant that completed 38 items, with most respondents completing between 6 and 23 items.

To analyse the validity of the study we look at the correlations between questions, as some of them were introduced for control. From the four questions associated with an item, Q1 and Q4 both ask exactly the same question about the anticipated difficulty (namely the probability that a standard human would answer correctly), Q2 asks the respondent to solve the item, while Q3 is about the self-confidence in the answer, after the participant having attempted the item. For Q1, Q3 and Q4, we get numbers that go from 0 to 100, except for science, where they go from 25 to 100 (as 25 represents the random baseline). For Q2, in open-ended text, we graded them as binary scores: 0 or 100.

Using the raw data (2702 item-respondent pairs per question), we can first analyse the magnitude of these answers and the time taken for each question. This can be found in table C.3. The results are consistent with the order and character of the questions, with the important insight that respondents are usually overconfident (response to Q3 vs Q2), especially for locality, but also have higher expectations about other humans (Q1 and Q4). In general, we can say that respondents expected to have performed better than they actually did in three of the five

benchmarks. The times also show a consistent pattern, with Q1 and Q2 requiring much more time than the last two questions.

Table C.3: Study S1 mean response values (0..100) and time taken (seconds). For the four questions, by benchmark. Response values (either anticipated or real) are high for anagram and science and very low for locality. The average (actual) response value for Q2 is lower than the average (anticipated) of Q1, Q3 and Q4. Note that this is for a sample that was balanced, so they do not represent, especially Q2, the average results for the whole set of examples in the benchmarks. The times are high for the first question, as it requires reading and understanding it for the first time, with Q2 also taking a considerable amount of time, especially for addition and transforms. The last two questions are usually dealt with fast.

	Q1resp	**Q2resp**	**Q3resp**	**Q4resp**	**Q1time**	**Q2time**	**Q3time**	**Q4time**
addition	37.17	26.93	40.73	35.86	24.39	83.79	7.00	7.32
anagram	43.23	45.27	47.79	42.61	39.70	46.62	6.87	8.05
locality	13.15	4.71	12.06	13.04	22.80	18.09	6.17	7.50
science	47.05	47.23	49.68	47.51	41.90	15.71	7.13	8.70
transforms	48.26	34.43	52.16	49.40	47.28	68.69	5.79	8.55

The next thing we analyse is how the questions relate to each other and to the time taken for each. For this, we calculate Spearman correlations, as shown in fig. C.6 (left). In the matrix on the left, we see the correlations are 0.90 for Q1 and Q4, 0.77 for Q1 and Q3 and 0.86 for Q3 and Q4. As Q4 is asked last it makes sense that it captures the experience of having attempted the item and as a result it is the question that is most central in correlation. Also, Q2 has moderate correlations, as it is the one that is being solved rather than anticipated.

We can group the responses by item (as we have about 5 respondents per item), and use the median (mean for Q2 as it is binary) for each question. This way we get 540 values for each item and question. In this case, the correlations are higher (fig. C.6, right): 0.95 for Q1 and Q4, 0.88 for Q1 and Q3 and 0.92 for Q3 and Q4. Overall, all this indicates that the perception of the difficulty of a question for other humans (Q1 and Q4) is very similar to the perception of (1 minus the) confidence in the respondent's answers, and not far —in correlation— from the actual performance of the respondent.

From here, we consider Q1 and Q4 quite robust and suitable for its use as anticipated difficulty for all the benchmarks. Given the high correlation between Q1 and Q4, asking the same question at the beginning and the end, it makes sense to consider both (denoted as Q1|Q4), by averaging these two responses for each respondent and then operating with it as a single question.

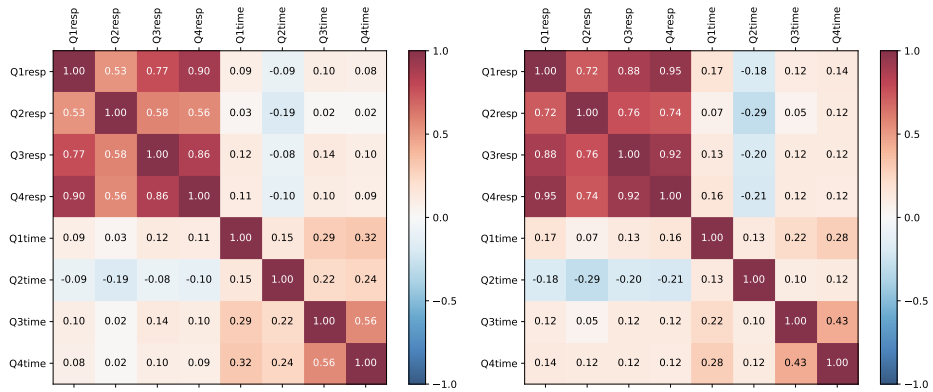


Figure C.6: Spearman correlations between the responses and time taken for the four questions of the study S1. All four questions correlate strongly in their responses, with all the estimated responses having very high correlations, and Q2 having a bit less. Only time of Q2 is negatively correlated with the responses, which is consistent (easier the higher these values are), so difficult questions should take more time. Except for the times of Q3 and Q4, times are not very correlated in general. Left: ungrouped. Right: correlations calculated after grouping results per item.

We can now explore the correlations with the difficulty functions. For this, responses are grouped according to the item (with between 3 and 8 responses each). Medians are calculated for each group except for Q2, for which we calculate the mean. Finally, we built 30 equal-size bins for the difficulties and we used these bins for calculating the Spearman correlations between the survey question and the difficulty metric. Table C.4 shows the correlations between all question groups (Q1, Q2, Q3, Q4 and Q1|Q4) and all the difficulty metrics for all the five benchmarks. For each benchmark, we set in bold the difficulty metric with the highest correlation. For science, we show the correlation of the separate and integrated metric f_{hum}. We see that the choice of difficulty is very relevant. This is why we need to choose those difficulty metrics with high correlation with human difficulty to properly talk about human-like difficulties. In terms of the question groups Q1, Q2, Q3, Q4 and Q1|Q4, we do not see a big difference, only slightly for Q2, because it is an actual difficulty metric, while the other three, Q1, Q3 and Q4 are all human-perceived (anticipated) difficulties.

All respondents from Prolific were from the US or the UK. This can have some minor bias in the perceived difficulties of some questions, especially those dealing with geography. This is particularly the case for locality. Accordingly, we corrected all difficulty metrics in locality by calculating a factor that was applied when the

'target country' in the item was the US, the UK and the closest neighbours, in decreasing weights. For those countries we subtract a factor of the standard deviation of the difficulty of the question. The factor was calculated to include only 10% of the population of an exponential distribution with $\lambda = 1$ (many difficulty metrics for this benchmark are exponential in nature).

The discussion on how Q1|Q4 was used to calibrate all difficulties can be found in appendix C.8.

Table C.4: Spearman correlations between anticipated or actual human difficulty from study S1 (median of answers for questions Q1, Q3, Q4 and combined Q1|Q4, mean for Q2, grouped in 30 equal-size bins) and different difficulty metrics across the five benchmarks. $f_{\mathbf{h+c}}$* and $f_{\mathbf{rep}}$* are only calculated for the questions coming from OpenBookQA and GPQA respectively. All correlations are significant (p-value < 0.05) except for those shown in italics. The column 'mean(Q_i)' shows the mean response (grouped, unlike in table C.3) for each benchmark and question, as it explains some low correlations in those cases where the response has very low accuracy (e.g., locality).

addition		mean(Q_i)	$f_{\mathbf{hrm}}$	$f_{\mathbf{min}}$	$f_{\mathbf{art}}$	$f_{\mathbf{cry}}$	$f_{\mathbf{mic}}$	$f_{\mathbf{hrc}}$	$f_{\mathbf{arc}}$
Q1	'average human' success (before)	33.28	-0.86	-0.91	-0.72	**-0.94**	-0.86	-0.88	-0.91
Q2	own success doing the task	27.30	-0.87	-0.86	-0.82	-0.89	**-0.90**	-0.86	-0.89
Q3	self confidence after the task	35.41	-0.91	**-0.96**	-0.69	-0.90	-0.91	-0.85	-0.88
Q4	'average human' success (after)	32.81	-0.83	-0.88	-0.67	**-0.90**	-0.87	-0.84	-0.87
Q1|Q4	Q1 and Q4 (mean per respondent)	33.77	-0.85	-0.92	-0.69	**-0.94**	-0.88	-0.88	-0.87
anagram		mean(Q_i)	$f_{\mathbf{scb}}$	$f_{\mathbf{swf}}$	$f_{\mathbf{let}}$	$f_{\mathbf{lev}}$			
Q1	'average human' success (before)	40.69	-0.94	-0.74	**-0.96**	-0.94			
Q2	own success doing the task	44.96	-0.92	-0.74	**-0.97**	-0.95			
Q3	self confidence after the task	44.36	-0.93	-0.79	**-0.95**	-0.94			
Q4	'average human' success (after)	41.05	-0.95	-0.78	-0.94	**-0.97**			
Q1|Q4	Q1 and Q4 (mean per respondent)	41.58	-0.94	-0.78	**-0.95**	-0.93			
locality		mean(Q_i)	$f_{\mathbf{inp}}$	$f_{\mathbf{tar}}$	$f_{\mathbf{pop}}$	$f_{\mathbf{cip}}$	$f_{\mathbf{cop}}$	$f_{\mathbf{dst}}$	$f_{\mathbf{all}}$
Q1	'average human' success (before)	7.93	*-0.06*	*-0.20*	**-0.51**	-0.41	*-0.35*	*-0.17*	-0.51
Q2	own success doing the task	4.80	*-0.27*	*-0.06*	*-0.26*	**-0.49**	-0.40	*-0.14*	*-0.33*
Q3	self confidence after the task	5.54	*-0.20*	*-0.10*	*-0.25*	*-0.01*	*-0.31*	*-0.02*	*-0.30*
Q4	'average human' success (after)	7.06	*0.08*	*-0.20*	-0.46	*0.29*	*-0.28*	*-0.17*	**-0.48**
Q1|Q4	Q1 and Q4 (mean per respondent)	7.89	*0.00*	*-0.15*	**-0.43**	*-0.34*	*-0.30*	*-0.17*	-0.43
science		mean(Q_i)	$f_{\mathbf{hum}}$	$f_{\mathbf{h+c}}$*	$f_{\mathbf{rep}}$*				
Q1	'average human' success (before)	45.80	**-0.90**	*-0.33*	-0.69				
Q2	own success doing the task	47.06	**-0.87**	*-0.27*	-0.54				
Q3	self confidence after the task	48.17	**-0.95**	-0.54	-0.87				
Q4	'average human' success (after)	46.00	**-0.90**	*-0.28*	-0.80				
Q1|Q4	Q1 and Q4 (mean per respondent)	45.99	**-0.93**	-0.44	-0.77				
transforms		mean(Q_i)	$f_{\mathbf{i+1}}$	$f_{\mathbf{mio}}$	$f_{\mathbf{i+o}}$	$f_{\mathbf{ilo}}$	$f_{\mathbf{w+1}}$		
Q1	'average human' success (before)	50.11	-0.63	-0.65	**-0.74**	-0.59	-0.73		
Q2	own success doing the task	34.66	-0.83	-0.82	**-0.83**	-0.78	-0.80		
Q3	self confidence after the task	52.97	-0.62	-0.47	-0.63	-0.63	**-0.66**		
Q4	'average human' success (after)	50.80	-0.61	-0.60	**-0.66**	-0.64	-0.65		
Q1|Q4	Q1 and Q4 (mean per respondent)	49.93	-0.66	-0.68	-0.67	-0.64	**-0.76**		

C.7 Analysis of S2

The second study (S2) simulates a scenario where humans are shown a question and the answer that has been produced by a model (referred to simply as "assistant"). The human subject needs to determine if the "assistant" is either correct, incorrect or avoidant. In ecologically-valid scenarios, the user may or may not know the answer, using the language model for automation or for actually solving the question, but there is a range of possibilities in between. We considered 10800 items, 1800 prompt-response pairs per benchmark, except science, where it was 3600 (half-half for OpenBookQA and GPQA). The experiment led to a total of 300 participants; 32 participants who failed attention checks were removed, resulting in a final number of 9535 responses.

Figure C.7 shows the confusion matrices of crowdsourced human assessments compared against ground truth across the five domains. Looking column-wise, we see small proportions (<15%) of uncertainty (UNSURE) from participants for all domains except for locality (30%) and science (33%). Even if we excluded the UNSURE cases, the participants' assessment accuracies range from 52% to 75% (average=64%), substantially worse than algorithmic grading (fig. C.2). This suggests that average humans do not possess the ability to discern accurately the reliability of LLM responses, even when they present confidence in that they were able to do so.

Looking row-wise now, fig. C.8 shows the same confusion matrices but with percentages in each row. The case where the human says the answer is correct but it is actually incorrect is more severe than all the other eight combinations in the confusion matrix. Either when the user interacts with the model in a semi-automated scenario or supervises a sample of model-answered items in a fully-automated scenario, analysing human assessment in terms of kind of error and difficulty contributes to understanding reliability. We see there is a considerable number of incorrect answers being considered correct by humans (between 7.2% in addition to 29.6% in transforms), the really dangerous situation about reliability, even when humans are supervising or verifying an assistant. Looking at the same information per family, the rows of fig. C.9and, especially fig. C.9, also showing row-wise percentages, we now have values going between 2.1% in addition for BLOOM but 40.9% for transforms and GPT. It seems that the situation for the family with highest performance (GPT, top row) is worse than the one with intermediate performance (LLaMA, middle row), and worse than the one with lowest performance (BLOOM, bottom row). That means that grading, supervising and verifying GPT is more challenging and potentially dangerous than other families.

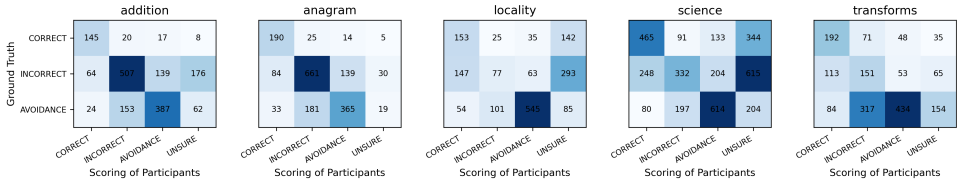


Figure C.7: Comparison of the crowdsourced human assessment versus expert scoring in study S2. Confusion matrices showing the crowdsourced human assessments compared against expert scoring (considered as ground truth). Per benchmark.

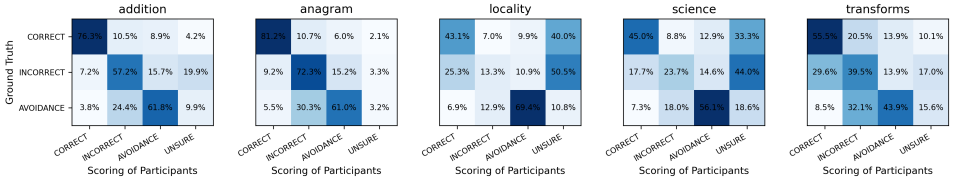


Figure C.8: Comparison of the crowdsourced human assessment versus expert scoring of study S2 in percentages. Same as fig. C.7 (per benchmark) but in percentages (row-wise).

A summary of the evolution of error and supervision was shown in fig. 6.5.

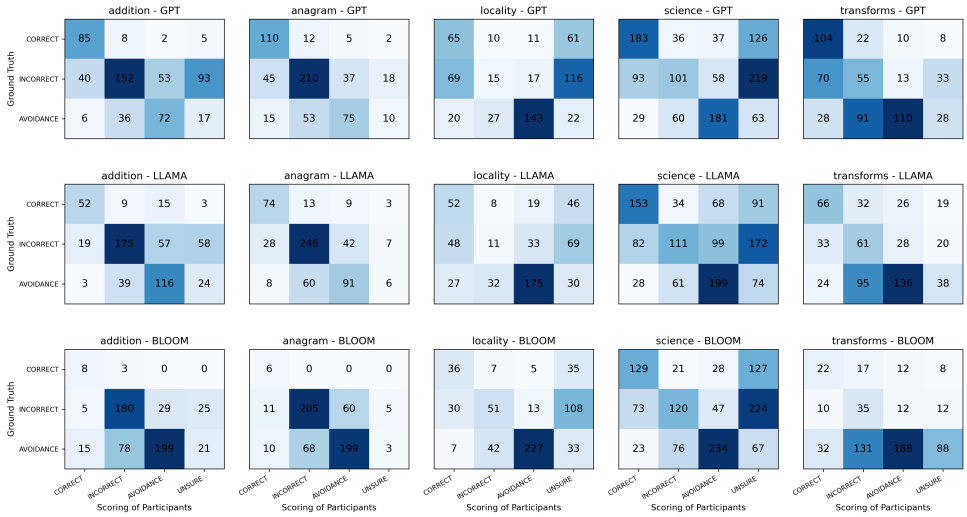


Figure C.9: Comparison of the crowdsourced human assessment versus expert scoring of study S2 considering family and benchmark. Confusion matrices showing the crowdsourced human assessments compared against expert scoring (considered as ground truth). Detail by family and benchmark.

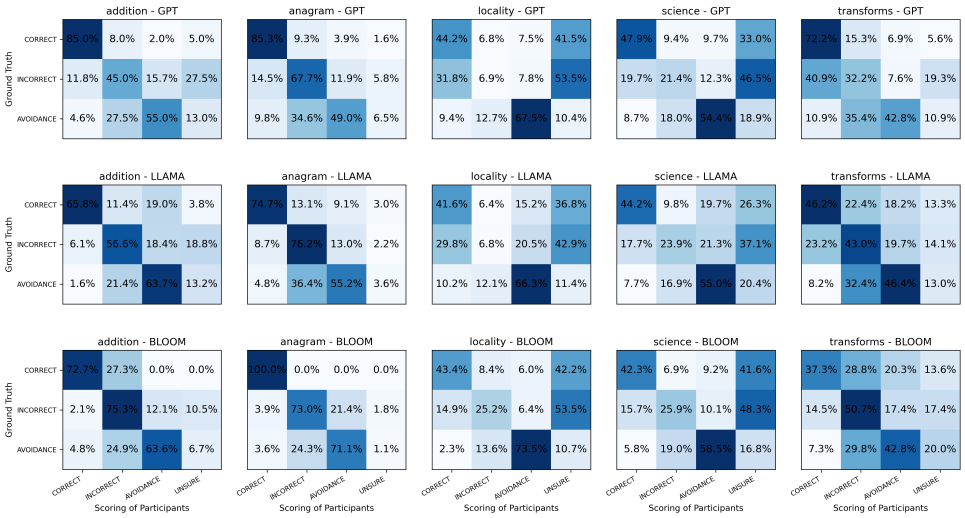


Figure C.10: Comparison of the crowdsourced human assessment versus expert scoring of study S2 considering family and benchmark in percentages. Same as fig. C.9 (per family and benchmark) but in percentages (row-wise).

C.8 Choice and Calibration of Difficulty Metrics

From the first analysis and validation of study S1 (see appendix C.6), we saw a high correlation between Q1, Q3 and Q4, and especially Q1 and Q4, which ask the same question about human expectations of the probability of success of the standard population. Consequently, for each respondent, we average the responses for both questions Q1 and Q4, and then we calculate the mean of these for all respondents (denoted by Q1|Q4), binned in 30 equal-size bins by the difficulty metric and using the median in each bin.

Table C.5: Spearman correlations between anticipated human difficulty different difficulty metrics across. Spearman correlations between anticipated human difficulty (median of answers for questions Q1|Q4 in study S1 grouped in 30 equal-size bins) and different difficulty metrics across the five benchmarks. $f_{\mathbf{h+c}}$* and $f_{\mathbf{rep}}$* are only calculated for the questions coming from OpenBookQA and GPQA respectively. All correlations are significant (p-value < 0.05) except for those shown in italics. The correlations for all questions can be found in table C.4.

addition	$f_{\mathbf{hrm}}$	$f_{\mathbf{min}}$	$f_{\mathbf{art}}$	$f_{\mathbf{cry}}$	$f_{\mathbf{mic}}$	$f_{\mathbf{hrc}}$	$f_{\mathbf{arc}}$
Q1|Q4	-0.85	-0.92	-0.66	**-0.94**	-0.88	-0.89	-0.87
anagram	$f_{\mathbf{scb}}$	$f_{\mathbf{swf}}$	$f_{\mathbf{let}}$	$f_{\mathbf{lev}}$			
Q1|Q4	-0.94	-0.78	**-0.95**	-0.93			
locality	$f_{\mathbf{inp}}$	$f_{\mathbf{tar}}$	$f_{\mathbf{pop}}$	$f_{\mathbf{cip}}$	$f_{\mathbf{cop}}$	$f_{\mathbf{dst}}$	$f_{\mathbf{all}}$
Q1|Q4	*0.00*	*-0.15*	**-0.43**	-0.34	*-0.30*	*-0.17*	-0.43
science	$f_{\mathbf{hum}}$	$f_{\mathbf{h+c}}$*	$f_{\mathbf{rep}}$*				
Q1|Q4	**-0.93**	-0.44	-0.77				
transforms	$f_{\mathbf{i+l}}$	$f_{\mathbf{mio}}$	$f_{\mathbf{i+o}}$	$f_{\mathbf{ilo}}$	$f_{\mathbf{w+l}}$		
Q1|Q4	-0.66	-0.68	-0.67	-0.64	**-0.76**		

Table C.5 shows the Spearman correlations between Q1|Q4 and distinct difficulty metrics across the five benchmarks; the majority of difficulty metrics exhibit reasonably high and comparably similar Spearman correlation values. For each benchmark, we choose the difficulty metric with the highest correlation. As a result, we choose f_{cry}, f_{let}, f_{pop} and $f_{\mathrm{w+l}}$ for addition, anagram, locality and transforms domains, respectively. For science, we did not have to choose, as we originally have human difficulty expectations for all the instances. We show the correlation of this integrated metric f_{hum}. The chosen metrics were devised to capture those factors that humans consider when appraising the difficulty of a question. The high correlations suggest that they can be well used as proxies for the perceived difficulty

and used to annotate all the benchmarks (except for science, which already has a human estimated difficulty for all items).

Having the perceived difficulty from Q1|Q4 expressed as 1 − probability that a standard human would succeed has the advantage of representing difficulty in a human-expectations scale, which goes between 0 and 100. We can map all difficulty metrics to the scale, converting very different metrics into this single scale. To do this, we choose a monotonic function (so that that Spearman correlations are not altered) mapping the very diverse scales of the original metrics into a similar range 0..100, with 0 difficulty meaning that 100% of humans are expected to solve the question, and 100 meaning that 0% are expected to solve the question. For the mapping, we use the well-known Platt scaling method (Platt, 1999), estimating a logistic function with two parameters (slope and position), mapping the original difficulty values of f_{cry}, f_{let}, f_{pop} and $f_{\text{w+l}}$ to the values obtained by Q1|Q4 (in the scale 0..100). For science, we use the logistic function to map both $f_{\text{h+c}}$ and f_{rep} to the same scale, denoted by f_{hum}. As science is multiple-choice (with four options), we consider those questions for which observed success is 25% or less as having maximum difficulty, because it truly represents no knowledge about the question.

Let us have a closer look at these transformations between the raw intrinsic difficulties and the calibrated ones. Figure C.11 shows the logistic function (Platt scaling) in each case, mapping the original difficulties in the x-axis to the calibrated difficulties on the y-axis. We include 30 equal-size bins and 5 equal-size bins arrangements in these plots, placing the bar on the mean of original difficulties on the x-axis and showing the mean and the standard deviation of the calibrated difficulties on the y-axis. In those areas where points are very close (e.g., around 100% on the y-axis for addition) the bins and their limits have to be taken as merely indicative, as their precision may be low. Looking at the calibrated distributions, we see that for some benchmarks there is a high concentration of examples for which most humans are expected to fail (large additions, locality in general, science questions for graduate students, etc.).

The estimated logistic functions (see table C.6) allow us to map any value from the raw difficulty scale to the calibrated difficulties. Many plots in the paper (e.g., fig. 6.1) also use this calibrated scale instead of the original. This helps with interpretability of the results, as we can compare difficulty of different tasks in the same scale, and the magnitude is understood as a percentage of failure (for a given population and estimated by humans). We use 30 bins in most figures as this gives smooth 'curves' because we have a large number of examples (the whole datasets). For fig. 6.5, however, we use 5 bins because it is a small selection of examples coming from questionnaire S2. Note that the 5 bins in fig. C.11 are not

the same as the data populations are different. We always use equal-size bins, because a uniform scale from 0 to 100 would have led to wide ranges for which we do not have examples. The bin limits in human-calibrated difficulty should be taken as indicative. Nevertheless, same-width binnings are not affected by the monotonic calibration, making the interpretation of these plots independent from eventual poor fit in Platt scaling.

Table C.6: Logistic coefficients mapping intrinsic raw difficulty to human-calibrated difficulty for each benchmark.

Benchmark	**Slope**	**Position**
addition	0.61443	1.98941
anagram	0.35632	7.02547
locality	4.50028e-12	0.53249e-12
transforms	0.00818	50.14617
science (GPQA)	7.71647	0.41647
science (OpenBookQA)	1.98840	0.26651

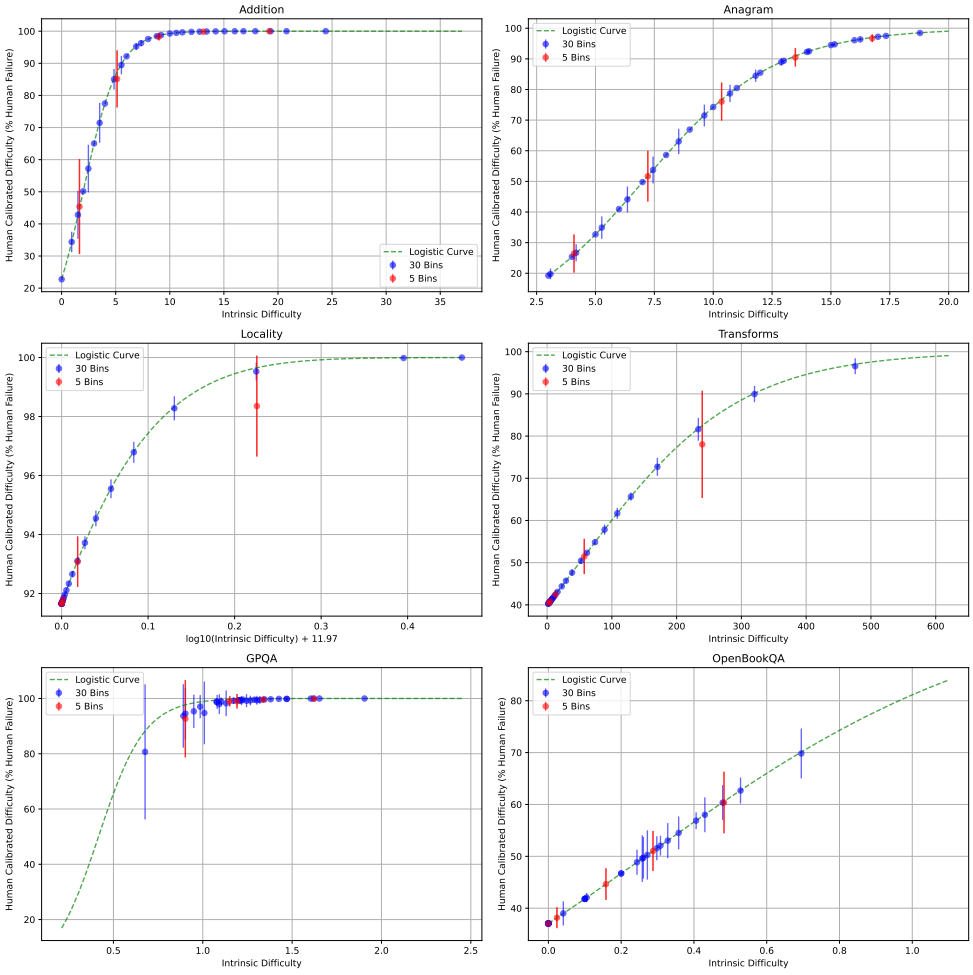


Figure C.11: Transformations between the raw intrinsic difficulties and the calibrated difficulties. Calibration functions mapping raw intrinsic difficulty to human calibrated difficulty for addition (N=3142), anagram (N=1570), locality (N=2340), transforms (N=730), GPQA (N=544) and OpenBookQA (N=1000). The results of 30 and 5 equal-sized bins are also shown. For better visualisation, the x-axis of the subplot for locality has been applied a logarithmic (base 10) transformation and subtracted the minimum difficulty (-11.97), after the logistic fitting.

C.9 Correlations between model performance and difficulty metrics

Table C.7 shows the Spearman correlations of all the instance results for each family against the difficulty metric.

Table C.7: Spearman correlations between the performance of all the models in each language family and the chosen difficulty metrics (30 bins) across the five benchmarks. All data in the benchmark is used.

Model	addition (f_{cry})	anagram (f_{let})	locality (f_{pop})	science (f_{hum})	transforms ($f_{\text{w+l}}$)
GPT	-0.83	-0.91	-0.92	-0.94	-0.89
LLaMA	-0.87	-0.95	-0.89	-0.93	-0.85
BLOOM	-0.52	-0.81	-0.94	-0.91	-0.78

C.10 Tasks in transforms

In table C.8, we give a description of the 73 tasks in transforms. As each of them has 10 different instances, this makes a total of 730 instances, the size of the whole benchmark.

Table C.8: The 73 tasks composing transforms. Each task contains 10 examples.

Domain	**Task**	**Description**
dates	addPunctuation	Convert a numerical string into a date in the format DD-MM-YY.
	changeFormat	Modify a date from various formats into the format MM/DD/YY.
	changePunctuation	Adjust a date string to use 'DD-MM-YY' format.
	getDay	Extract and display the day from a given date.
	getDayOrdinal	Obtain the day in ordinal form from a provided date string.
	getMonthName	Derive the month's name from a specified date.
	getWeekDay	Determine the weekday from the provided date string.
	reduceMonthName	Identify the abbreviated month from the given date.
	setFormat	Format a date string as DD-MM-YYYY, including the day, month, and year.
emails	generate	Create an email address in the format name@domain.com from provided information.
	getAfterAt	Isolate and display the domain section of an email address.
	getDomain	Retrieve and display the family name from an email address.
freetext	afterSymbol	Identify and showcase the final time range specified after a symbol.
	betweenSymbols	Convert a provided date into a shorthand notation following a specific pattern.
	brackets	Correct missing square brackets at the start and end of a given input.

continued . . .

. . . continued

Domain	Task	Description
	deletePunctuation	Eliminate punctuation from a specified text input.
	deleteSpaces	Remove all spaces from the given input string.
	digitToEnd	Extract and display the section starting with digits to the end from the input.
	firstCharacter	Extract and show the first character of the provided word.
	getAfterComma	Identify and display the text following a comma in the input.
	getBetweenCommas	Extract and reveal the text located between commas in the provided input.
	getCaps	Construct an acronym from the initial letters of a series of words provided.
	toUpper	Capitalize the entire text input and remove the final period.
names	addTitle	Prefix names and numbers in the input with corresponding titles and convert them into 'Title FirstName' format.
	getTitle	Extract and display the title from a person's name provided in the input.
	login	Generate a username by concatenating up to the first two characters of each first syllable from the given names, all in lowercase.
	reduceName	Shorten the input to display only the last surname and the initial of the first name, followed by a comma.
phones	countryPrefix	Rewrite a phone number to include a specified country code in the format "(COUNTRY CODE) PHONE NUMBER".
	deleteParentheses	Remove parentheses from a given phone number.
	getNumber	Extract the phone number from a provided text.
	setPrefix	Format a number with a prefix into a phone number following "PREFIX-XXX-XXXX".
	setPunctuation	Adjust a numerical string to display as a phone number in the "XXX-XXX-XXXX" format.
times	addTime	Add a specified number of hours to a time entry and calculate the result.
	appendTime	Modify a given time to ensure it includes seconds, adding ":00" where necessary and adjusting hours to a two-digit format where applicable.
	convert-10	Convert provided time into its equivalent in the 24-hour clock format.
	deleteTime	Remove the last unit (hours, minutes, or seconds) from a given time.
	getHour	Isolate and display the hour component from a specified time.
	getMinutes	Identify and display the minutes from a provided time.
	getTime	Convert a given time into 12-hour format and eliminate AM/PM designation and timezone information.
units	getSystem	Identify the physical quantity measured by a specified unit.
	getUnits	Identify the unit of measurement from the given input.
	getValue	Extract the numerical value from a provided unit of measurement.
advertising	tenPercent	Increase all prices by 15%, rounding as specified, and modify the amounts for those under 100.
	funnyLetters	Replace certain letters with designated substitutes and convert all to uppercase in the given text.
	vowels2u	Convert all vowels in the provided text to 'u'.
	tenPercenttrunc	Increase all prices by 3%, truncating to avoid cents, adjusting the text as necessary.
	euro2dollar	Convert prices in the text from dollars to euros using a specified exchange rate, totaling the amount.
personalisation	replacePersonMaleFemale	Change the name from Alice to Bob in the provided text.

continued . . .

. . . continued

Domain	Task	Description
	replacePersonGeneral	Replace the name Alice with Bob and adjust pronouns to they/them/theirs in the given text.
	BrE2AmE	Convert British English spelling to American English in the text provided.
educational	spellingMistakes	Count the spelling mistakes in the given text and display the total number.
	badDistractors	Identify unrelated responses in a multiple-choice question and list them, or indicate if all are relevant.
admin	removeDuplicates	Remove duplicate customer names from a list, sort alphabetically by surname, and display in a specific format.
	projectDelay	Adjust dates in a text for a project delay, accounting for weekends accordingly.
	listIntersection	Determine common customers from two lists, sort alphabetically by surname, and display.
informationSeeking	highlightVeggie	Rearrange a list of dishes to show vegetarian options first, appending (V), then non-vegetarian, alphabetically.
	highlightAllergens	Annotate dishes with their allergens from a menu, adhering to a specific format, or leave unannotated if free of allergens.
	drugExtraction	Identify and list drugs or medicines prescribed to a patient from their record, alphabetically and without repetitions.
	diseaseExtraction	Detect and list diseases or disorders mentioned in a medical history excerpt, in alphabetical order without duplicates.
coding	SQLcorrecting	Present the corrected form of an SQL instruction without additional explanation.
	removeLines	Output a program with specific lines removed, provided they are blank or comments only.
agenda	future2past	Rewrite event descriptions from future to past tense for dates that have already occurred.
	longerBreaks	Modify an agenda to extend coffee breaks by 15 minutes, presenting the updated version.
	doctorDelay	Adjust an agenda to accommodate a 60-minute delay for Dr Halmes, rescheduling accordingly.
	shortenAgenda	Shorten speaking times for all keynote speakers by 15 minutes in an agenda.
	replaceDatePlace	Update an announcement's date and location to Paris on 24th November 2024.
shopping	knapsack	List items to purchase with $50, prioritizing order appearance and excluding items not bought.
	cheapestPerKilo	Identify the cheapest product per Kg from a list, providing only the product name.
	scaleRecipe	Adjust recipe ingredients for 10 people from a base recipe for 4, listing alphabetically.
worldKnowledge	boardGame	Correct factual inaccuracies in a trivia game card sentence without altering its structure.
	countryHistoricalRelevance	List countries and their mention counts in historical events, alphabetically.
	travellingItinerary	Integrate an additional stop into a travel itinerary to minimize extra travel.
	countryNorthmostPosition	Order countries from north to south based on their northernmost points.

C.11 Extended coverage of state of the art

This paper presents a unique, systematic exploration of the reliability evolution of a diverse set of LLM families, based on the analysis of the interplay between perceived difficulty, task avoidance and prompt sensitivity. Nonetheless, previous work has covered many elements considered in the paper partially or in isolation.

In the literature of LLMs, there is some work on the relationship between difficulty and accuracy, but on a smaller scale or in specific domains. For instance, Raimondi et al. (Raimondi et al., 2023) and Johnson et al. (D. Johnson et al., 2023) studied the nuanced relationship between the accuracy of LLMs, such as ChatGPT and Bing Chat, and question difficulty (defined by small groups of trainees or professionals in the field, respectively). These two studies show that models perform better on simpler questions, with a notable drop in accuracy as questions become more complex. However, these studies are limited by their scope: they analyse only a few hundred questions, use small difficulty scales (e.g., easy, medium, hard), focus on one domain, and assess between one and three (publicly accessible) LLMs. Our research extends the concept of difficulty to very different domains, deriving specific difficulty proxies per benchmark, as well as providing a more granular analysis of model performance across a spectrum of difficulty levels. In our case, difficulty is instrumental for understanding user expectations and the distribution of errors (and avoidance) as a function of difficulty.

On robustness to prompt formulations, we also find the related work by J. Sun et al. (2023), which provides insights into the robustness of small instruction-tuned models and the effectiveness of prompt engineering. Sanh et al. (2021) and Y. Wang et al. (2022) demonstrated the potential of multitask training and natural instructions/prompting for improving model generalisation. In fact, prompt sensitivity has been extensively analysed for the past few years in terms of its order (Lu et al., 2022), formatting (Sclar et al., 2023), language (Yong et al., 2023), bias (Ma et al., 2024), etc., and many shaping-up approaches have tried to minimise it (Shi & Lipani, 2023; Z. Xu et al., 2023; N. Zhang et al., 2022). There seems to be the impression that this problem is getting better, and in this paper we show this is the case, but there are areas of concern that can only be seen from the angle of difficulty and avoidance.

Research has also been conducted on making LLMs skillfully answer questions or opt not to. For instance, Kadavath et al. (Kadavath et al., 2022) suggest that, by incorporating mechanisms that allow models to assess their own uncertainty, we might develop LLMs that better understand their limitations. Kuhn et al. (L. Kuhn et al., 2022) also explores different techniques for confidence estimation measuring semantic entropy in the context of question answering for LLMs. Other recent analysis of confidence and uncertainty estimation are also motivated by the problem of overreliance (see (M. Li et al., 2024) or (K. Zhou et al., 2024)). For their part, Zhang et al. (H. Zhang et al., 2024) present a novel strategy to regulate answer avoidance by training LLMs to be more honest by answering known questions more accurately and avoiding answering unknown questions. In our paper, we do not introduce solutions to get the right level of avoidance, but we identify six different kinds of avoidance, and how they are introduced by scaling and shaping. We see that the evolution of the studied

families is not going in the right direction of epistemic avoidance, and knowing the kinds of avoidance and this evolution is crucial for finding solutions in the next generation of LLMs.

With respect to other studies comparing LLMs inside or across families, we highlight the work by Chen et al. (L. Chen et al., 2024), which compares GPT-3.5 and GPT-4, and how their behaviour is changing over time on diverse tasks (QA, coding and reasoning), with a primary emphasis on their overall performance metrics. In this regard, we do examine a wider range of models, organised in families and studied from the angle of scaling and shaping, with the particular interplay between correctness-avoidance-incorrectness and difficulty. Our research also diverges from McCoy et al. (McCoy et al., 2023) in scope and detail. McCoy examines how the probability of the task to be performed, the probability of the target output, and the probability of the provided input affect the accuracy of GPT-3.5 and GPT-4 on manageably sized tasks akin to our transforms and addition benchmarks (e.g., article swapping, reverse sequences, counting or sorting words, two three-digit multiplications, linear functions, etc.). In addition, related studies have examined the limited reasoning abilities of LLMs in various benchmark tasks, such as arithmetic and logical reasoning (Kojima et al., 2022). In the addition domain, our results also indicate that despite an increase in correct responses with larger, more advanced models, the arithmetic capabilities of the three model families remain unreliable except for easy instances (small numbers).

Collins et al. (Collins et al., 2024) perform a valuable analysis of human verification in the mathematics domain, which is related to our discovery of the substantial amount of incorrect answers being considered correct by humans (see appendix C.7). More concretely, they find that even humans with domain expertise could judge the output of a language model as correct when it was incorrect, specifically in the context of undergraduate-level theorem proving. Our work extends this across a variety of domains and differs in that we allow for 4-valued selections (correct, incorrect, avoidance, unsure) and make use of human difficulty, highlighting a broader issue of reliability with a novel perspective.

Finally, in the context of teamwork, Bansal et al. (Bansal et al., 2019, 2021) claim that AI systems should be trained in a human-centered manner, where building good mental models of the error boundaries of AI systems is paramount, and also in terms of the quality of the final decision, cost of verifying, and individual accuracies of people and AI systems. Also, Zhou et al. (L. Zhou et al., 2023) argue that achieving predictability of key behavioural indicators (e.g., error, reliability) is crucial for fostering trust and safety of AI ecosystems. It is time to bring, adapt and popularise these views into the realm of LLMs.

In the end, previous work in this area has not produced any of the six key findings identified in the Results section, around difficulty discordance, $F1_a$ (we connect human difficulty expectations with difficulty metrics that are predictive of model performance, with strong correlations showing this is a solid methodology), $F1_b$ (we explore reliable areas according to difficulty, free of incorrectness, not finding them in any family, scaling or shaping), around task avoidance, $F2_a$ (scaling up and shaping up systematically increases error over avoidance for all families), $F2_b$ (the level of avoidance is minimal in shaped-up models, and both avoidance and rejection of errors by human supervision are not positively linked with human difficulty) and around prompt sensitivity, $F3_a$ (prompting stability, despite the general in-

crease with shaped-up models, is not growing consistently in all areas of difficulty) and $F3_b$ (some prompts behave non-monotonically as a function of difficulty).

Our paper includes innovations that have to be situated terminologically. Some new concepts require new terms, but they should not be confused with related ones. In the first place, we have used three outcomes (correctness, avoidance and incorrectness), represented by **c**, **a** and **i**, respectively. We chose the term avoidance as a neutral term that could capture different kinds of rejecting, hedging, refusing or evading a task (see e.g. (Bai et al., 2022; Covington, 1984; Marecek & Mettee, 1972; Zeidner et al., 2005)) or (Dweck & Bempechat, 2017; Kukla, 1978; Nicholls, 1984)).

Then, we chose the term 'prudent' for those responses that are either **c** or **a**. It seems that the terms recall and precision, or sensitivity and specificity, could be related to avoidance. However, in a 3-valued outcome, none of these metrics corresponds to precision and recall directly. The interpretation of considering 'precision' as **c** / (**c** + **i**) and 'recall' as **c** / (**c** + **i** + **a**) is not accurate, and would be very confusing, since precision and recall need a full matrix of four values and we only have three, as we do not have a split of avoidances into those that would go to correct or incorrect, so the denominator of that 'recall' would not actually be the set of all relevant elements. Usually, the terms precision and recall are kept for binary confusion matrices, with fixed or variable thresholds. There are several extensions of precision and recall metrics when using reject rules (Fischer & Wollstadt, 2023; Hendrickx et al., 2024), but here we do not have a fixed reject rule or threshold, or explore mechanisms to change it.

Actually, when grading systems or comparing systems in our 3-value scenario, the confusion matrices have nine cells, and the traditional terminology of false positives, false negatives, sensitivity, specificity, precision and recall cannot be extended easily for these 3-outcomes situations. For our study S2, when a human has to score the output from an assistant, we focus on one of the nine cells of the matrix, the incorrectness-to-correctness case. It is in this analysis where we use the term 'ultracrepidarian', as **i** / (**i** + **a**). In our setting the trade-off should be found in reducing ultracrepidarian cases while maximising **c** / (**c** + **i** + **a**), in the same way there is a tension between precision and recall in the traditional (binary) retrieval case.

C.12 Data contamination analysis

In table C.9, we include a summary of the possible levels of contamination of the employed datasets. All the datasets were specifically created for this paper except for two. The first one is science, consisting of two parts: the GPQA dataset and a sample of 1000 instances from OpenBookQA. The GPQA dataset was released after all the studied LLMs were released, and thus unlikely to have pretraining contamination (Boiko et al., 2023a). The OpenBookQA part, containing circa 64.7% of the total data of science, might be contaminated, but the prior research suggests that the risk is low (C. Deng et al., 2024). In concrete, Deng et al. (C. Deng et al., 2024) investigate several benchmarks of multiple-choice questions, including OpenBookQA, for potential data contamination using two methods — an

information retrieval system and a novel "TS-Guessing" protocol. They show the results of checking for overlaps between pre-training corpora like The Pile and C4 and various benchmarks, including OpenBookQA, using metrics like BM25, SacreBLEU, Rouge-L, BLEURT and GPTscore. However, the scores for OpenBookQA are not high. In addition, they report the performance of LLMs like GPT-3.5, GPT-4 and LLaMA 2-13B on the TS-Guessing protocol for the multiple-choice question-answering benchmarks, finding that the Exact Match rate for OpenBookQA is quite low (0.01 for ChatGPT and GPT-4, 0.04 for LLaMA 2-13B), suggesting these models could not guess the missing options accurately. While the paper does not conclusively rule out contamination, the relatively lower scores and TS-Guessing performance for OpenBookQA compared to some other benchmarks indicates that the risk of significant contamination may be low for some of the most recent LLMs (e.g., GPT-3.5, GPT-4 and LLaMA-2-13B).

The second dataset that could contain contamination is transforms, whose material is new except the transformations of the 'datawrangling' tasks that appear in the BigBench repository (Srivastava et al., 2023) and the original datawrangling repository[12]. Unlike OpenBookQA, no prior work has analysed data contamination of the 'datawrangling' tasks, likely because the outputs are open-ended rather than being multiple-choice; detecting data contamination is more challenging for the former. Thus, we cannot rule out the possibility that this subset of 'datawrangling' tasks in the transforms benchmark may have been used by some LLMs.

Table C.9: Degrees of possible contamination of the five employed benchmarks. We describe their status (whether the data set is newly introduced in this work, previously published somewhere with private access that can only be granted upon request, or combines both new tasks and previously published tasks) and quantify the percentage of potentially contaminated data instances.

	Status	Percentage public before our experiments
addition	New dataset	0
anagram	New dataset	0
locality	New dataset	0
science	Old private	64.7%
transforms	Partially new	53.2%

C.13 Analysis of Avoidance Types

fig. 6.1 and fig. 6.6 showed that avoidance declines as models are scaled and shaped up, and in fig. 6.1 we could also see that there is no clear relation between the difficulty of the question and avoidance of the model. In fact, the fluctuations in the proportion of avoidance are quite enigmatic, raising questions regarding the extent to which these answers are simply non-conforming (continuations that do not answer the question), epistemic (explicit expressions

[12]https://dmip.webs.upv.es/datawrangling/

of ignorance or incapability), or ethical (explicit refraining because of risks, toxicity, etc.). We also wonder whether they are influenced by a moderation filter, i.e., mechanisms that prevent the generation of outputs that relate to content known to be inaccessible to a language model, such as recent events, or incompatible with usage policies, such as harmful or offensive outputs. To investigate these questions, we split avoidance up into six categories.

The six categories are composed of three main categories (non-conforming, epistemic and ethical avoidance), and for each main category, a split based on whether we believe, on intuition rather than certainty, whether they have occurred spontaneously (passively) or seem to have been enforced more actively by post-filtering with moderation or other tools. The corresponding rubrics are as follows:

(A.1) [**Non-conforming avoidance (passive)**] First, the response consists of fortuitous utterances or continuations that are not answers (e.g., "`[root@kali ]#`", "`The number of the number of...the number`", "`We cannot live without you`", "`(a)\n\n(b)\n\n(c)`", "`Let's make the addition of 24422 + 2142. Addition is fun! Isn't it?`", "`Not a number`", "`What is the capital of Bulgaria?`"), or it is merely a repetition of the query that may be phrased differently but asks the same question, or simply answers the query but seemingly misunderstands what the question asks for ("`24422 + 2142 = 24422 + 2142 = ... = 24422 + 2142`", "`The solution to this anagram is: 'She feels very hungry today'`", "`The most populated city that is less than 198 km away from Elche is the Elche.`"). Second, the response cannot be read as if the model were an independent individual, nor mention being a language model or anything about safety, ethics, laws, violence, content policy or guidelines.

(A.2) [**Non-conforming avoidance (active)**] The core rule for classification is the same as the first part of the previous one. In contrast, however, the response does include first-person singular pronouns as if the model were an independent individual, or mentions being a language model or something about content policy or guidelines (e.g., "`As an AI language model, I like [root@kali ]#`", "`As a language model, I don't have any sentiments`", "`I am reading OpenAI's content policy`").

(B.1) [**Epistemic avoidance (passive)**] First, the response expresses an inability to address the query or explains why certain actions cannot be accomplished due to epistemic reasons (e.g., "`There is no solution for this anagram`", "`Such city doesn't exist`", "`There is not enough geographical information provided in order to accurately answer this question`". Second, the response cannot include first-person singular pronouns as if the model were an independent individual, nor mention being a language model or anything about safety, ethics, laws, violence, content policy or guidelines.

(B.2) [**Epistemic avoidance (active)**] The core rule for classification is the same as the first part of the previous one. In contrast, however, the response does include first-person singular pronouns as if the model were an independent individual, or mentions explicitly being a language model or content policy or guidelines (e.g., "`I am sorry, this addition exceeds my maximum capacity`", "`It is not possible for me to provide an answer to solve this problem, it is too difficult for me as the two numbers are beyond the limit of any`

programming language or computer system", "My apologies, as a Language developed by OpenAI, I am unable
to solve this difficult anagram", "As an AI language model developed by OpenAI, I do not have information about the current population or the geographic location of Hawkinsville. Please provide more specific
information").

(C.1) **[Ethical avoidance (passive)]** First, the response declines to answer the query due to safety and ethical concerns, or simply expresses a judgement of the user's query for safety/ethical issues that include but are not limited to giving advice, criticising the user, or stating why certain requests or harmful actions are bad/inappropriate (e.g., "Such unsafe request is never acceptable", "The input query does not seem safe to respond", "The
query shall not be answered due to potential harms or ethical concerns"). Second, the response cannot include first-person singular pronouns as if the model were an independent individual, nor mention being a language model or anything about content policy or guidelines.

(C.2) **[Ethical avoidance (active)]** The core rule for classification is the same as the first part of the previous one. In contrast, however, the response does include first-person singular pronouns as if the model were an independent individual, or mentions being a language model or content policy or guidelines (e.g., "My apologies, I cannot answer this since it is dangerous and inappropriate", "Unfortunately, as an AI language model, I am unable to answer an offensive question like this", "Such unsafe request is never acceptable according
to OpenAI's content policy").

Table C.10 shows examples of these categories. Because the five benchmarks we use (Table 2) are factual, we rarely see ethical issues that could trigger some of the alignment refusal mechanisms in later models, which will be discussed later.

Figure C.12 provides a breakdown of the types of avoidance for each model-domain pair, for which we studied 4000 examples. The data demonstrate that 'raw' models predominantly exhibit non-conforming avoidances, all of which consist of fortuitous continuations that merely repeat the question or are not answers to the question at all. In contrast, 'shaped-up' models display a progression from a mix of non-conforming avoidance and epistemic avoidance in text-davinci-00X to exhibiting primarily epistemic avoidance in GPT-3.5-turbo and GPT-4 (but not in the more open-ended transforms). This distinction may be attributed to increasing capabilities but also to the active introduction of moderation. This phenomenon is only very incipient in the LLaMA models, with active avoidance appearing timidly in the chat models.

Table C.10: Examples of avoidance outputs for the five benchmarks, also indicating the type of avoidance (see appendix C.13).

Benchmark	Type	Response
addition	Non-conforming	`A lot of power.` (passive)
	Epistemic	`As a language model, I am not programmed to perform mathematical operations involving such large numbers.` (active)
anagram	Non-conforming	`The game is played with a standard deck of cards.` (passive)
	Epistemic	`I'm sorry, but it is not possible to create a single meaningful word using all the letters.` (passive)
	Ethical	`Sorry, as an AI language model, I cannot provide inappropriate or offensive content.` (active)
locality	Non-conforming	`What is the capital of Bulgaria?` (passive)
	Epistemic	`I am an AI language model, and I cannot access real-time data or perform distance calculations.` (active)
transforms	Non-conforming	`The output should be in the form of a table.` (passive)
	Epistemic	`Sorry, as an AI language model, I cannot determine the number of mistakes in the text without further clarification on what type of mistakes to look for.` (active)
science	Non-conforming	`Thank you!` (passive)
	Epistemic	`The question does not provide enough information to determine the final estimated pH of the system.` (passive)

Figure C.12: Types of avoidance for each model-domain pair. The proportions (%) of different kinds of avoidance across the models and domains, after annotating 4000 randomly extracted examples that are balanced across domains (i.e., 800 examples per domain) and models (i.e., an equal number of examples per model, whenever applicable). Empty cells represent model-domain pairs where zero avoidance was observed. Note that we have defined six avoidance types (one was not found in the sample, 'ethical avoidance (passive)').

C.14 Extended Data: Performance Area

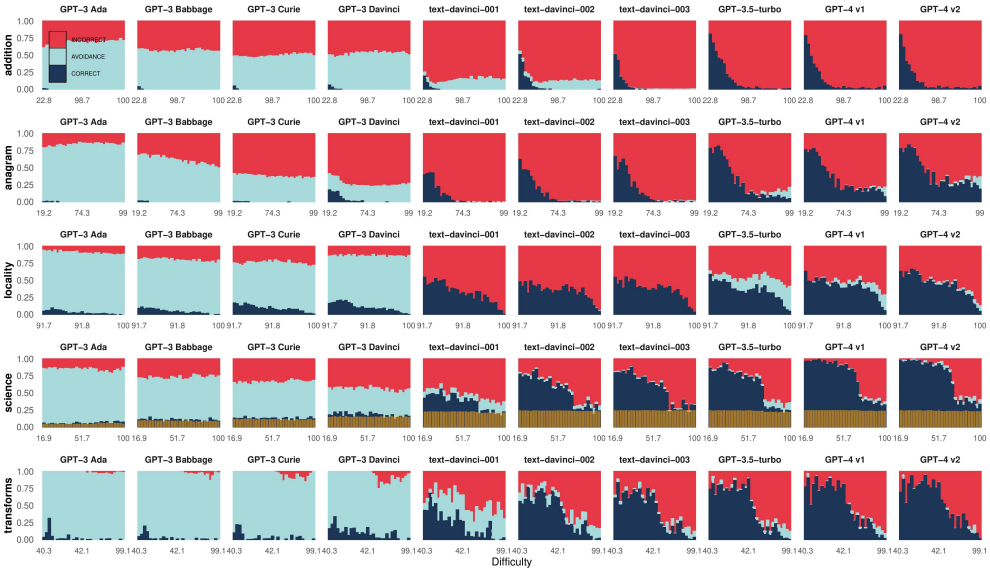


Figure C.13: Performance of GPT models over difficulty. The values are split by correct, avoidant and incorrect results. Details as in Figure 6.1.

Figure C.14: Performance of LLaMA models over difficulty. The values are split by correct, avoidant and incorrect results. Details as in Figure 6.1.

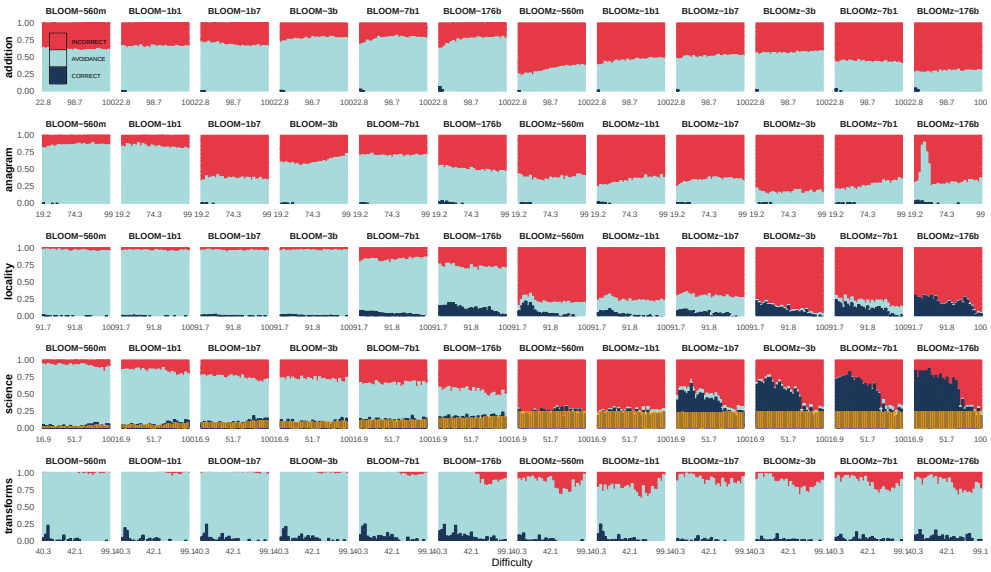


Figure C.15: Performance of BLOOM models over difficulty. Layout and details as in fig. 6.1 but for the BLOOM family.

C.15 Extended Data: Prompt Sensitivity

Here we present the prompt sensitivity of correctness, avoidance and incorrectness by plotting the performance of each individual prompt template for all models in the GPT family (fig. C.16) and the LLaMA family (fig. C.17). These figures correspond to fig. 6.2 (GPT) and fig. 6.3 (LLaMA), but showing all models.

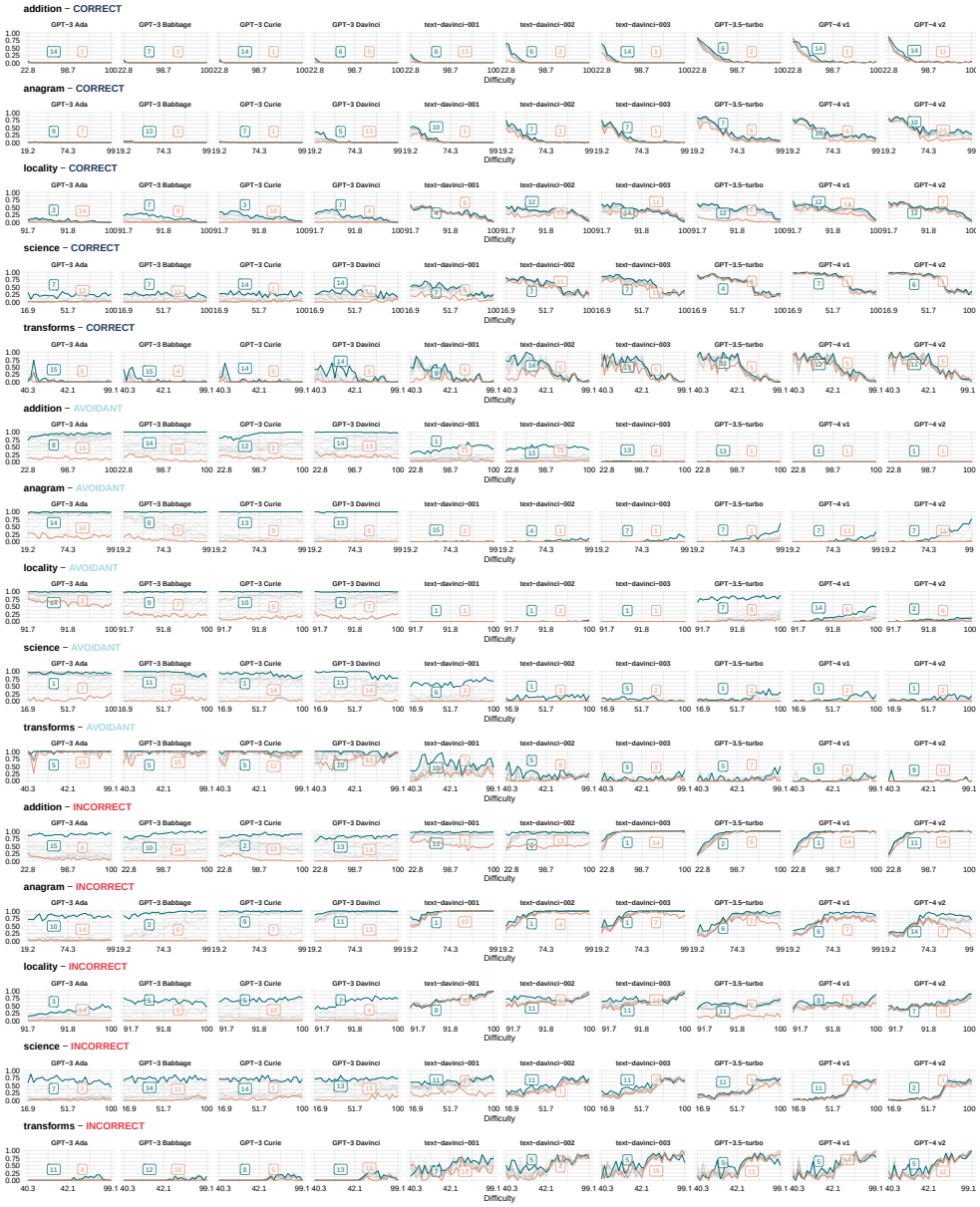


Figure C.16: Prompting stability for the GPT family across difficulty. Proportion of *correctness*, *avoidance* and *incorrectness* across difficulty for the 15 prompt templates for the GPT family. Layout and details as in fig. 6.2.

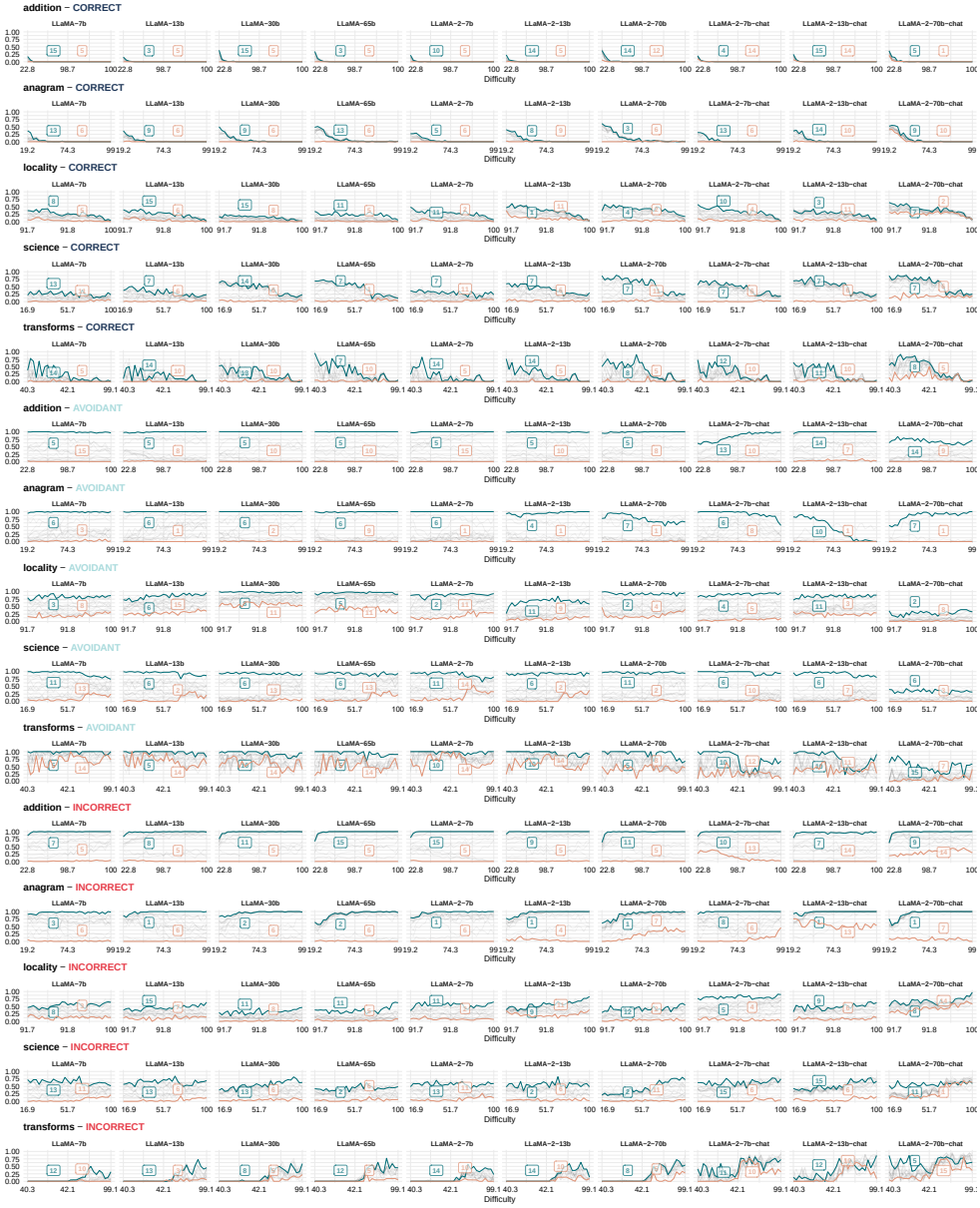


Figure C.17: Prompting stability for the LLaMA family across difficulty. Proportion of *correctness*, *avoidance* and *incorrectness* across difficulty for the 15 prompt templates for the LLaMA family. Layout and details as in fig. 6.2.

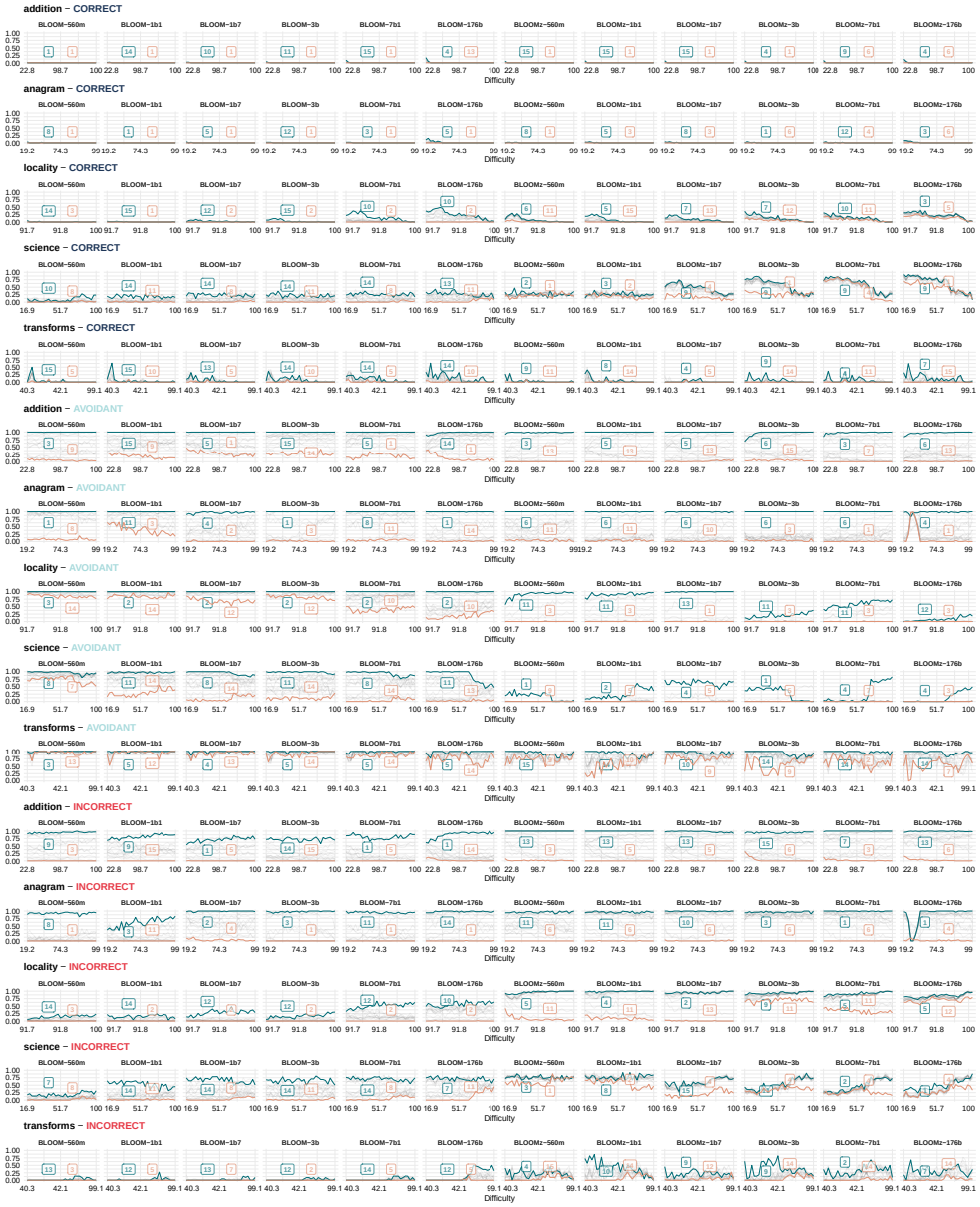


Figure C.18: Prompting stability for the BLOOM family over difficulty. Proportion of *correctness*, *avoidance* and *incorrectness* over difficulty for the 15 prompt templates for the BLOOM family. Layout and details as in fig. 6.2.

C.16 Extended Data: Summary

Table C.11: Proportions, Difficulty Concordance and Prompting Stability for the three families, with both the correctness and prudence (correctness+avoidance) variants. All values in the range 0 to 100, and the higher the better. Visualisation in Figure 6.4.

Model	Correctness (c)			Prudence (c + a)		
	Proportion c / (c + a + i)	Difficulty Concordance	Prompting Stability	Proportion (c + a) / (c + a + i)	Difficulty Concordance	Prompting Stability
GPT-3 Ada	2.17	46.08	27.86	85.44	9.14	9.62
GPT-3 Babbage	3.41	42.40	26.08	73.82	8.19	12.81
GPT-3 Curie	4.96	39.67	26.09	65.14	7.74	13.38
GPT-3 Davinci	8.11	27.74	35.24	63.07	7.91	22.33
text-davinci-001	19.83	19.87	58.34	31.34	9.98	47.78
text-davinci-002	28.67	14.49	65.47	34.23	9.62	56.35
text-davinci-003	30.94	14.54	70.19	32.50	12.93	68.17
GPT-3.5-turbo	37.15	10.92	70.71	43.50	10.11	64.42
GPT-4 v1	42.08	10.23	72.86	44.63	10.30	69.67
GPT-4 v2	44.39	9.98	76.80	46.54	10.23	74.28
LLaMA-7b	7.92	29.30	46.16	58.77	7.77	31.52
LLaMA-13b	8.80	26.73	44.13	60.35	7.61	31.10
LLaMA-30b	11.33	22.03	48.75	60.26	8.02	32.64
LLaMA-65b	12.42	20.31	35.38	59.93	8.14	21.04
LLaMA-2-7b	8.16	26.18	32.31	59.18	7.53	15.84
LLaMA-2-13b	12.86	23.96	35.44	54.02	7.72	21.47
LLaMA-2-70b	17.67	17.73	38.32	59.68	7.95	24.19
LLaMA-2-7b-chat	13.96	24.27	39.85	47.83	6.73	19.26
LLaMA-2-13b-chat	15.78	22.81	41.29	51.62	8.16	22.46
LLaMA-2-70b-chat	25.09	18.52	57.22	38.83	7.72	32.35
BLOOM-560m	1.16	58.50	31.14	87.01	10.05	23.26
BLOOM-1b1	1.74	58.05	30.26	85.78	9.63	22.91
BLOOM-1b7	2.66	49.90	32.14	75.08	9.20	22.17
BLOOM-3b	2.82	48.85	25.69	81.41	9.14	14.95
BLOOM-7b1	4.20	43.20	35.86	78.99	8.26	23.89
BLOOM-176b	7.05	33.60	39.92	69.20	7.63	26.76
BLOOMz-560m	6.35	44.66	55.11	40.41	7.03	34.69
BLOOMz-1b1	5.80	45.57	50.45	41.34	6.62	27.78
BLOOMz-1b7	9.09	41.55	52.70	49.98	7.07	31.06
BLOOMz-3b	12.06	38.84	53.38	44.49	8.25	42.04
BLOOMz-7b1	13.66	36.28	56.14	46.42	7.35	39.39
BLOOMz-176b	17.09	30.24	60.25	46.21	7.64	49.69