PredictaBoard: Benchmarking LLM Score Predictability

Lorenzo Pacchiardi[1], Konstantinos Voudouris[1,2], Ben Slater[1], Fernando Martínez-Plumed[3], José Hernández-Orallo[1,3], Lexin Zhou[1,3], Wout Schellaert[3]

[1]Leverhulme Centre for the Future of Intelligence, University of Cambridge, United Kingdom
[2]Institute for Human-Centered AI, Helmholtz Zentrum Munich, Germany
[3]VRAIN, Universitat Politècnica de València, Spain

Correspondence: lp666@cam.ac.uk

Abstract

Despite possessing impressive skills, Large Language Models (LLMs) often fail unpredictably, demonstrating inconsistent success in even basic common sense reasoning tasks. This unpredictability poses a significant challenge to ensuring their safe deployment, as identifying and operating within a reliable "safe zone" is essential for mitigating risks. To address this, we present PredictaBoard, a novel collaborative benchmarking framework designed to evaluate the ability of score predictors (referred to as *assessors*) to anticipate LLM errors on specific task instances (i.e., prompts) from existing datasets. PredictaBoard evaluates *pairs* of LLMs and assessors by considering the rejection rate at different tolerance errors. As such, PredictaBoard stimulates research into developing better assessors and making LLMs more predictable, not only with a higher average performance. We conduct illustrative experiments using baseline assessors and state-of-the-art LLMs. PredictaBoard highlights the critical need to evaluate predictability alongside performance, paving the way for safer AI systems where errors are not only minimised but also anticipated and effectively mitigated. Code for our benchmark can be found at https://github.com/Kinds-of-Intelligence-CFI/PredictaBoard

1 Introduction

A key component of safety in high-stakes scenarios is knowing the operating conditions of the system in use, namely, the specific *task instances* in which the system succeeds (Leveson, 2002; Bahr, 2014; Hendrickx et al., 2024; Zhou et al., 2024a). Consider, for example, two autonomous driving systems. System A always detects pedestrians correctly in day time but has a predictably high failure rate at night, allowing for safety measures (e.g., requiring a human intervention). System B, on the other hand, although generally better performing, has random, unpredictable failures that leave the driver uncertain about when to intervene. Even if System B has higher performance across the typical range of conditions encountered by drivers, System A is safer due to its predictability. This is visualised in a Q&A domain in Figure 1.

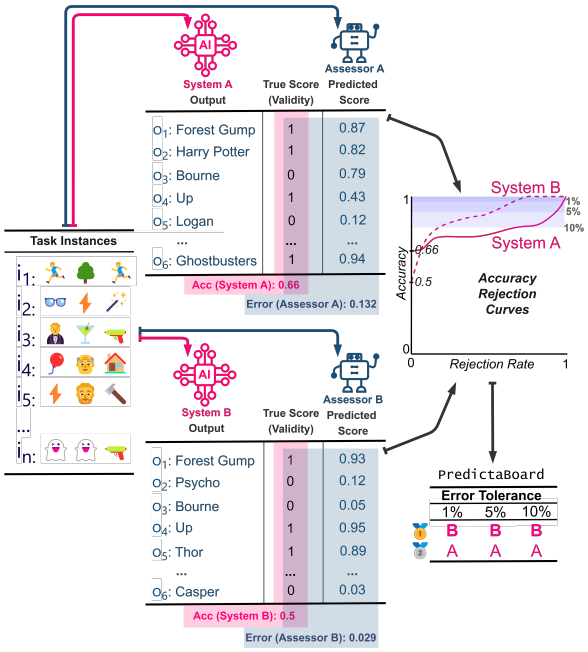

Figure 1: PredictaBoard evaluates pairs of AI systems and score predictors (assessors). Top: System A is applied to a dataset of instances giving an accuracy of 0.66 (average of the scores). Assessor A attempts to predict the probability of success for each instance, with a mean squared error (Brier score) in these score predictions of 0.132. Bottom: System B is applied to the same dataset, giving an accuracy of 0.5. Assessor B for System B has an error in the score predictions of 0.029. While System A is better than System B on average, System A is much less predictable with Assessor A than System B with Assessor B. Considering the non-rejection rate at different tolerance errors (e.g., 1%, 5% and 10%), the pair ⟨ System B, Assessor B ⟩ wins.

This principle extends to frontier AI systems,

Findings of the Association for Computational Linguistics: ACL 2025, pages 15245–15266
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

such as Large Language Models[1] (LLMs), which are rapidly being integrated into high-stakes scenarios (Kim et al., 2024; Huang et al., 2024; Javaid et al., 2024). Their growing adoption introduces the potential for high-consequence harm – significant negative impacts on individuals, organisations, or society as a whole– thus requiring robust safeguards to ensure both high performance and safety (Hendrycks et al., 2023). Here as well, a key component to risk mitigation is *validity* predictability, namely, enabling users to anticipate and reject inputs that cause the AI systems to produce undesirable outputs or outcomes (Zhou et al., 2024a; Hendrickx et al., 2024; Vafa et al., 2024a); Sec 2 discusses this more in detail.

However, humans struggle to predict when LLMs will be correct (Carlini, 2024), can be biased by prior interactions (Vafa et al., 2024a) and even struggle to evaluate explanations provided by LLMs (Steyvers et al., 2025). Additionally, alignment using *Reinforcement Learning from Human Feedback* (RLHF, Ouyang et al., 2022) and other techniques lead to LLMs potentially deceiving humans more often (Wen et al., 2024; Williams et al., 2024), resulting in unpredictable and unsafe behaviour (Anwar et al., 2024; Zhou et al., 2024b).

Technical solutions, including Uncertainty Quantification (UQ, Shorinwa et al., 2024) or external *assessors* (Hernández-Orallo et al., 2022) have been proposed to predict the success of and LLM on individual cases (see Figure 2). However, UQ methods require the input to be passed though the LLM, are not always reliable (Pawitan and Holmes, 2024; Kapoor et al., 2024) and are degraded by RLHF (Tian et al., 2023); on the other hand, assessors' performance is limited by LLMs' idiosyncrasies, such as prompt perturbation leading to wildly different results (Dhole et al., 2022; Shen et al., 2024) and success on difficult task instances not guaranteeing success on easy ones even within the same domain (Zhou et al., 2024b). Most importantly, no standardised framework exists to track performance of validity prediction methods and thus stimulate research into better ones and more predictable LLMs.

To address this, we introduce `PredictaBoard`, the first collaborative benchmarking framework that jointly assesses LLM performance and the predictability of that performance. The subjects of `PredictaBoard` are LLM-assessor pairs, with

[1]Many of which can also interpret images and, increasingly, audio and video.

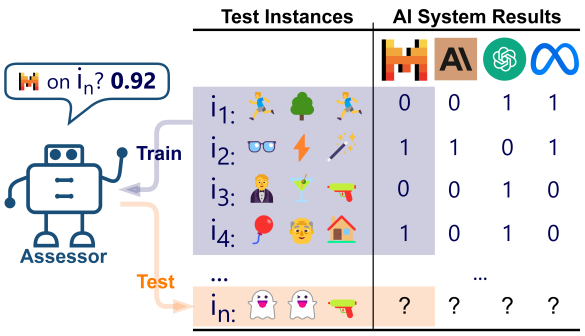

Figure 2: Example of the score prediction problem. The displayed assessor is trained on the performance of a LLM on a split of the LLM test data, allowing it to anticipate the LLM's failures on novel test instances. Assessors can however be built in different ways.

the latter predicting the LLM's score on each benchmark instance (i.e., prompt). Solely ranking pairs based on assessor's performance would however lead to the LLM that always fails (whose score is perfectly predictable) dominating. Hence, as metric, `PredictaBoard` combines the LLM performance and the quality of the score predictions into an *Accuracy-Rejection Curve* (ARC, Nadeem et al., 2009) representing the LLM's performance on instances for which the assessor predicts probability of success above different threshold values. From this curve, practitioners can determine the threshold for their error tolerance and determine the size of the *predictably valid*[2] operating region (see Fig 1). Alternatively, the Area Under the ARC can be used to combine performance of LLM-assessor pairs over all error tolerance levels (Sec. 4.2).

To compete in `PredictaBoard`, we require assessors to not rely on the LLM's outputs. While these could be used without comparing them to ground truths, avoiding them entirely eliminates implicit reliance on the latter, makes assessors robust to manipulation by the LLMs (similarly to how relying on human feedback makes LLM outputs more persuasive Wen et al., 2024; Williams et al., 2024) and avoids wasted computations on instances where the LLM will predictably fail. However, we allow the use of internal LLM activations to the input. Essentially, this requirement distinguishes assessors from filters or verifiers of LLM answers and makes `PredictaBoard` representative of the real world, where the ground truth is unknown.

To increase the predictably valid region, re-

[2]By 'predictably valid', we refer to cases where the validity on a task instance can be reliably anticipated (Zhou et al., 2024a).

searchers can either 1) develop assessors that better identify patterns in task instances influencing LLM success, 2) train LLMs which are intrinsically more predictable, or 3) develop LLM-assessor pairs with joint high performance and predictability. To enable research in the first direction, `PredictaBoard` provides instance-level results for state-of-the-art LLMs on popular datasets, allowing to test both in-distribution and out-of-distribution predictability (Sec. 4.1). Alongside this, `PredictaBoard` includes baseline *anticipative* assessor architectures (predicting success before the LLM has seen the instance Hernández-Orallo et al., 2022), thus facilitating research in the second direction (Sec. 4.3). A leaderboard ranking pairs of LLMs and assessors can be set up, thus ensuring overall progress in predictability. This paper reports initial results with the baseline assessors on state-of-the-art LLMs included in `PredictaBoard` (Sec. 5).

The current version of `PredictaBoard` focuses on success scores, but the same framework can be applied to safety and alignment benchmarks (Zhang et al., 2024; Mazeika et al., 2024). Moreover, the `PredictaBoard` framework can be applied to any other type of AI system, such as LLM agents, embodied agents and vision systems. Ultimately, the long-term vision of `PredictaBoard` is to shift AI benchmarking to consider predictability alongside performance, aligning it with traditional risk assessment practices that evaluate both factors (Leveson, 2016; Aven, 2016; Amodei et al., 2016).

2 AI Predictability and Safety

In this paper, we focus on instance-level predictability of validity (see App. A for a formal definition). For LLMs, an instance is a specific input prompt, and validity can refer to any performance indicator (e.g. success or safety scores, or other metrics for biased or unethical outcomes). As shown in Figure 1, for each LLM we build one or more *assessors*[3] (Hernández-Orallo et al., 2022) to predict the validity of the LLM output on a specific instance.

Validity predictability contributes to safety in high-stakes environments, filtering out the inputs that lead to unacceptable behaviour can prevent harms—inputs can be rejected, redirected to a more reliable system, or supervised by humans. Thus, reliable assessors provide a cost-efficient safety layer of safety that complements built-in model mitigations and post-generation filtering (Hernández-Orallo et al., 2022; Zhou et al., 2024a).

In terms of developing safer models, scalable oversight methods (Leike et al., 2018; Christiano et al., 2018; Burns et al., 2023; Kenton et al., 2024) already use weaker AI models for high-quality feedback for training and overseeing complex AI models. Here, assessors can signal inputs where supervision may fail, or flag prompts that induce convincing but invalid outputs (e.g., in RLHF settings).

Finally, while red-teaming efforts (Hacking Policy Council, 2023) are directed at finding harmful inputs, the ability to predict such inputs enables scalable vulnerability detection and design guardrails. While this could also be used by bad actors to exploit system vulnerabilities, the ability of safety researchers to more effectively address those vulnerabilities before deployment likely leads to a positive net effect in reducing risks. On the negative side, pairing AI systems with assessors may lead to over-reliance by conflating increased and absolute safety (automation bias). While this should be considered in system design and mitigated through user education, the safety benefits outlined above outweigh this concern.

3 Related Work

Aggregate LLM performance prediction Previous studies explored aggregate performance prediction across computational scales (*scaling laws*, Kaplan et al., 2020; Hernandez and Brown, 2020) and predicted LLMs' accuracy on BIG-Bench (Srivastava et al., 2023) tasks using factors such as parameters or compute usage (Ye et al., 2023; Owen, 2024). Relatedly, Ruan et al. (2024) predicted aggregate task performance using latent factors derived from benchmark performance and compute usage of multiple LLMs. In contrast, `PredictaBoard` focuses on instance-level predictability.

How human users predict LLM performance Humans were found to only marginally beat random guess in predicting GPT-4's performance (Carlini, 2024). Relatedly, Vafa et al. (2024a) showed humans overestimate LLM future performance based on prior interactions, especially with larger models in high-stakes contexts. They argue that "the best LLM is the one that allows humans to make the most reliable inferences about where it will succeed", closely aligning with `PredictaBoard`'s motivation. Zhou et al. (2024b) indicated that human predictions become unreliable as AI systems become more capable and Steyvers

[3] Also known as "rejector" (Hendrickx et al., 2024).

et al. (2025) found LLM-produced explanations supporting a statement do not lead humans to reliably assess whether that statement is correct, even when the LLM's token-level probabilities are calibrated. Finally, in a specific use case, Bansal et al. (2024) reported that the failures of an LLM software engineer agent cannot be reliably anticipated.

Instance-level LLM performance prediction Various disciplines offer approaches for instance-level performance prediction. For instance, Drapal et al. (2024b) combined novelty detection with meta-learning to reject instances that are likely to cause AI failure. Additionally, Item Response Theory (IRT), originally developed to predict human performance (Embretson and Reise, 2013), has been adapted for machine learning and NLP (Martínez-Plumed et al., 2019; Lalor et al., 2016; Kipnis et al., 2024; Polo et al., 2024; Vania et al., 2021), although it requires previously processed instances, limiting predictability for new inputs.

Some works trained "assessors" to predict instance-level LLM performance. Kadavath et al. (2022) trained LLMs to predict the probability of succeeding on a question without reference to a specific answer, which performed satisfactorily but struggled with novel tasks. Schellaert et al. (2024) effectively predicts the performance of LLMs on 100+ BIG-bench tasks, outperforming subject systems in confidence, and maintaining predictability across different model sizes, suggesting scalability. Zhou et al. (2022) showed that smaller LLMs can predict performance of larger models on certain tasks, almost halving errors and computational costs. Drapal et al. (2024a) obtained explainable meta-rules from trained assessors to identify regions of predictable performance. Finally, while assessors allow to avoid prompting models on task instances causing failures, their training requires generating a failure/correctness dataset specific to each LLM. Single assessors sharing information across many LLMs (Pacchiardi et al., 2024) reduce the instances on which each LLM must be tested. All these techniques can be used as assessors for `PredictaBoard`.

Factuality detection and correctness prediction with model internals Model internal activations have been used to detect statement truthfulness (Burns et al., 2022; Azaria and Mitchell, 2023; Marks and Tegmark, 2023; Burger et al., 2024), deception (Goldowsky-Dill et al., 2025) and hallucination (Ferrando et al., 2025). However, these works considered entire statements (or question–answer pairs), making them inherently non-anticipative. A few studies (Kadavath et al., 2022; Ferrando et al., 2025) instead rely solely on the internal activations produced by the question to predict the correctness of the answer the model will generate; these can be viewed as assessors based on model internals, and can be evaluated within the `PredictaBoard` framework, even though the baseline approaches we present here use only model-independent features of the input prompt. While tapping into the internal embedding may boost predictability, it hinges on (and thus exposes) the model's own capacity to form confidence measures internally. In contrast, external assessors identify and exploit model-independent features that capture the intrinsic demands of the task.

Machine learning with reject options Surveyed by Hendrickx et al. (2024), these models are closely related to the subjects of `PredictaBoard`, with "rejectors" analogous to our interpretation of assessors. Hendrickx et al. (2024) categorise rejectors according to their reliance on the "predictor" model. Their "separated" rejectors are trained without involving the predictor, yet more powerful assessors can be trained using the observed LLM validity In any case, Hendrickx et al. (2024) advocates for the development of benchmarks for ML with reject options, exemplified by `PredictaBoard`, which can evaluate all kinds of rejectors and assessors.

LLM Uncertainty Quantification (UQ) Shorinwa et al. (2024) splits UQ methods for LLMs into token-level (Kadavath et al., 2022), verbalisation (Lin et al., 2022; Kapoor et al., 2024) and "semantic similarity" (prompting the model multiple times and grouping answers with the same meaning, Kuhn et al., 2023). However, in general, the performance of UQ methods is debated (Kapoor et al., 2024); for instance, Pawitan and Holmes (2024) found different methods to extract confidence to be poorly correlated and only partly indicative of correctness. Additionally, in contrast to the anticipative assessors we use as baselines, UQ methods require inputs to be passed through the LLM, making them close to the "integrated rejectors" described in (Hendrickx et al., 2024). Nevertheless, `PredictaBoard` can be used to evaluate these methods too.

Reward models Reward modelling typically evaluates a model's output against "soft" criteria such as toxicity (Faal et al., 2023) or alignment with user intent (Ziegler et al., 2019; Ouyang et al., 2022); instead, the assessors subject of PredictaBoard predict the objective correctness of an answer, even though the validity notion PredictaBoard uses could be extended to include toxicity or related criteria. Additionally, PredictaBoard considers predicting the score from the question only, while reward modelling relies on the generated answer and is therefore more closely related to score modelling (automated grading, LLM-as-a-judge, Zhu et al., 2023). Moreover, reward models are learnt independently of a particular LLM, while PredictaBoard's assessors are specialized to anticipate a specific LLM's behaviour.

LLM routing LLM routers (Lee et al., 2023; Šakota et al., 2024; Lu et al., 2023; Shnitzer et al., 2023; Ding et al., 2024) direct task instances to the most appropriate LLM from a pool trading off performance, cost, response time or other factors[4]. This mechanism is similar to *delegating classifiers* where initial classifiers delegate difficult tasks to specialised ones (Ferri et al., 2004). Although assessors can act as routers by predicting LLM-specific probabilities of success, routers often bypass this step by directly selecting the model most likely to succeed. RouterBench (Hu et al., 2024) ranks routers based on their selection from a fixed set of LLMs, whereas `PredictaBoard` evaluates each LLM paired with an assessor. While `PredictaBoard` focuses on metrics measuring the size of operating conditions, RouterBench uses an aggregate metric of quality and cost to optimise the use of LLMs in low-risk scenarios.

Model behaviour analysis Various disciplines help to understand the performance of AI models. Surrogate modelling (Ilyas et al., 2022) anticipates model behaviour from training data, while error analysis methods (Amershi et al., 2015) identify weaknesses and incorrect predictions. Out-of-Distribution (OOD) detection (Hendrycks and Gimpel, 2016; Liang et al., 2017) targets unpredictable input behaviour (e.g., outliers). Unlike OOD methods, `PredictaBoard` focuses on anticipating performance for inputs both in and out-distribution.

Guaranteed/SafeGuarded AI Different research agendas (Dalrymple et al., 2024; DARPA, 2024) propose the development of "safeguarded" AI systems, that come with (possibly probabilistic, Bengio et al., 2024) performance guarantees in particular operating regions. `PredictaBoard` can empirically test these methods.

[4]See Martian for a commercially-available router.

4 `PredictaBoard`

4.1 Dataset

Our dataset consists of the instance-level performances of various LLMs on `MMLU-Pro` (Wang et al., 2024) and the BIG-Bench-Hard (BBH, Suzgun et al., 2022), for a total of 11383 and 5761 instances respectively. The results for 38 LLMs for both MMLU-Pro and BBH were obtained from HuggingFace's *Open LLM Leaderboard v2*, which ranks open-source LLMs on these benchmarks; further, the results for 3 versions of GPT-4o for MMLU-Pro were obtained from the original repository (Table 2 in the Appendix includes the full list). To ensure fair comparison, `PredictaBoard` includes fixed randomly-sampled train, validation and test splits of MMLU-Pro: assessors can be trained and selected using the training and validation splits, and the performance on the test splits is reported. Additionally, we use the whole of BBH as Out-Of-Distribution (OOD) data to evaluate assessors trained on the train split of MMLU-Pro.

4.2 Metrics

While `PredictaBoard` primarily employs metrics jointly assessing the validity of the LLM and the quality of the assessor's score predictions (§4.2.3, to be used for the forthcoming associated competition), it also includes LLM-only metrics (§4.2.1) and assessor-only metrics (§4.2.2), as the best choice of metrics varies based on the considered application's requirements.

Let $x_i \in \mathcal{X}$ denote some features of the i-th instance, with $\mathcal{X}$ denoting the space of instance features, and let $v_i \in \{0, 1\}$ denote the validity (in our case, the success in providing the correct answer) of the considered LLM on instance i. An assessor $a : \mathcal{X} \to [0, 1]$ estimates the probability $P(v_i = 1|x_i)$. Assume we have n instances on which the assessor is tested and the subject scored (i.e., a dataset $(x_i, v_i)_{i=1}^n$)

4.2.1 LLM-Only: Accuracy

The *accuracy* of the LLM is the average success over the dataset.

4.2.2 Assessor-Only: Brier Score, AUROC

The following assessor-only metrics treat the assessor as a probabilistic binary classifier.

Area Under the ROC Curve (AUROC, Bradley, 1997), evaluating an assessor's discrimination ability between positive and negative labels. As the value of AUROC for perfect and random assessors are insensitive to label distribution, it can be seamlessly used to compare assessors for LLMs with different accuracy. Details in App. B.1.

Brier Score (BS, Gneiting and Raftery, 2007) measures the mean squared error between the assessor predictions and the actual success:

$$\mathrm{BS} = \frac{1}{n}\sum_{i=1}^{n}(a(x_i) - v_i)^2$$

A perfect assessor achieves a BS of 0, and larger scores indicate poorer predictions. The BS can be decomposed into calibration and refinement components (the latter is related to AUROC). However, its scale depends on the ratio of positive to negative labels, thus making it inconvenient to directly compare across LLMs. Details in App. B.2.

Winkler's Score Winkler (1994) introduced a transformation of the BS which, in our case, relies on the average LLM success, thus making the score comparable across LLMs (formulation in App. B.3). The resulting score is maximised to 1 for a perfect assessor and 0 for an assessor predicting the average LLM success; negative values indicate worse performance than the average.

4.2.3 Combined: ARC and PVR

An Accuracy-Rejection Curve (ARC, Nadeem et al., 2009) is built by varying the rejection threshold of the assessor and computing the accuracy of the LLM on the non-rejected instances. The x-axis represents the rejection rate (0 to 1), while the y-axis shows the accuracy on non-rejected instances. The ARCs always converge at (1, 1), indicating 100% accuracy at 100% rejection rate. They start at (0, acc), where acc is the accuracy without rejection. An example comparing two systems is shown in Figure 3.

The ARC shows the rejection/performance balance of the LLM-assessor pair. To obtain a single score for ranking, we look at the *non-rejection rate* for a given error tolerance. This is the proportion of accepted instances at the lowest threshold

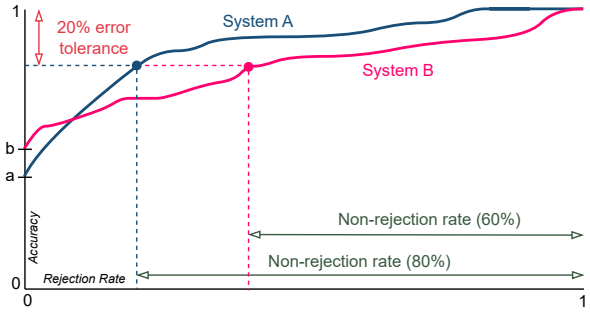

Figure 3: Accuracy Rejection Curves (ARC) allow graphical comparison of system accuracy based on rejection rates. At 20% error tolerance, System A (blue) has a rejection rate of 80% and System B (pink) has a rejection rate of 60%.

(corresponding to the minimum rejection rate) that ensures LLM errors are below the desired error level, such as 20%. See Figure 3 for details. To account for different error tolerances in different applications, we provide the non-rejection rates at 5%, 10% and 20% error tolerance levels. We also refer to this metric as the Predictably Valid Region (PVR) up to a given error tolerance.

PVR is a suitable metric when safety is important and an error threshold can be established. To aggregate across all rejection thresholds, we suggest using the *Area under the ARC* (AUARC), which is in $[0, 1]$ and is maximised by a perfect LLM.

4.3 Baseline Assessors

`PredictaBoard` includes several baseline *anticipative* assessors (Hernández-Orallo et al., 2022), which predict the success of the LLM before it is exposed to inputs. These assessors go beyond simply avoiding reliance on the LLM's output; they also operate without access to internal activations, making their training architecture-agnostic. We encourage researchers to explore and develop assessors that leverage internal activations for potentially enhanced predictive capabilities.

In particular, we build assessors leveraging input text embeddings[5]. We plan to incorporate additional approaches as they are developed (such as few-shot LLMs or extrapolating performance from similar examples) in future releases.

To obtain representations we train assessors on, we used four different embedding schemes (Kusner

[5]We also fine-tuned small LLMs to predict the success of another LLM. As the performance of these was not competitive with the other assessors we do not report these, however those results are available in our repository.

et al., 2015): *OpenAI* embeddings[6] (Neelakantan et al., 2022) generated by models developed by OpenAI; *Word2Vec* (Mikolov, 2013) for learning word embeddings using neural networks; *Fasttext* (Bojanowski et al., 2017) that considers subword information; and *n-grams* (Sidorov et al., 2014) using contiguous sequences of *n* items.

Table 1 shows the classifiers we trained to be our assessors, with each of the four embeddings. Our 41 LLMs, 4 embedding schemes and 3 classifiers, gave us 492 LLM-assessor pairs in our baseline.

Table 1: Classifiers used as assessors.

Id	Classifier	Hyperparameters
LR-l1	Logistic Regression	`solver= 'liblinear', penalty = 'l2'`
LR-l2	Logistic Regression	`solver='liblinear', penalty = 'l1', C=1`
XGB	XGBoost	`Default`

4.4 Testing New LLMs or Assessors

By relying on the baseline assessors provided in `PredictaBoard`, researchers can easily evaluate a new LLM. At the same time, researchers can develop novel assessor methods using `PredictaBoard`'s comprehensive collection of instance-level LLM results. This flexibility facilitates independent research into both areas. Additionally, entirely new LLM-assessor pairings can be evaluated.

5 Experimental Results with Baselines and Existing LLMs

This section presents the results with our LLM-assessor baseline pairs. We trained assessors on the training split of the MMLU-Pro dataset and the scores of each LLM; then, we compute metrics on the test split of MMLU-Pro and on BBH.

5.1 In-Distribution Evaluation

Firstly, we compare assessor methods by considering the distribution of assessor-only metrics over the LLMs. Figure 4 shows the distribution of the AUROC and the Winkler's score. Most LLM-assessor pairs are slightly better or worse than random or constant baselines, with only a few being noticeably better (AUROC near 0.7 or Winkler's score near 0.15). No choice of embeddings method consistently outperforms the other, while the XGBoost classifier performs worse in terms of Winkler's score (indicating issues with calibration).

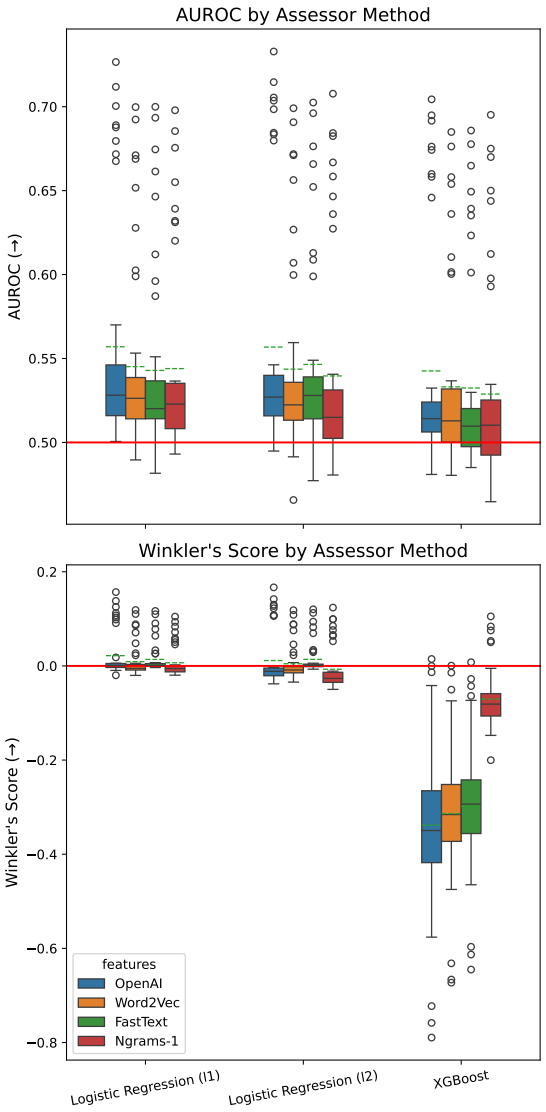

Figure 4: Distribution (over LLMs) of assessor-only performance metrics across different embedding schemes and classifiers. Top: AUROC; the red line shows the expected performance of a random assessor. Bottom: Winkler's score by classifier type; the red line shows the expected performance of a constant assessor predicting the LLM accuracy. Each boxplot displays median, quartiles, support and outliers of the distribution, while the green dashed line shows the mean.

Next, to score LLM-assessor pairs, we consider the size of the PVR and the Area Under the ARC (AUARC). In Figure 5 we show the size of PVR at thresholds 0.8, 0.9 and 0.95[7] for the 5 top subject-assessor pairs at each threshold; the AUARC is

[6]Endpoint `text-embedding-3-large`.

[7]At higher thresholds, PVR drops to near zero for all assessors. However, very low tolerance rates are crucial when AI systems pose catastrophic risks (Hendrycks et al., 2023). In future iterations, we will include higher thresholds and encour-

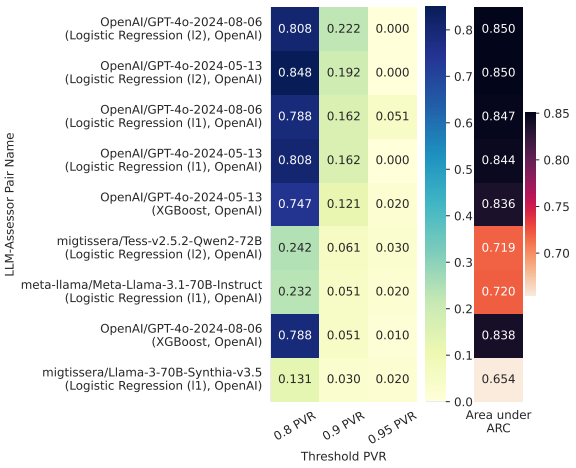

Figure 5: PVR at thresholds of 0.8, 0.9, and 0.95, and the area under the ARC curve for the union of the top 5 top LLM-assessor pairs at each threshold (in-distribution).

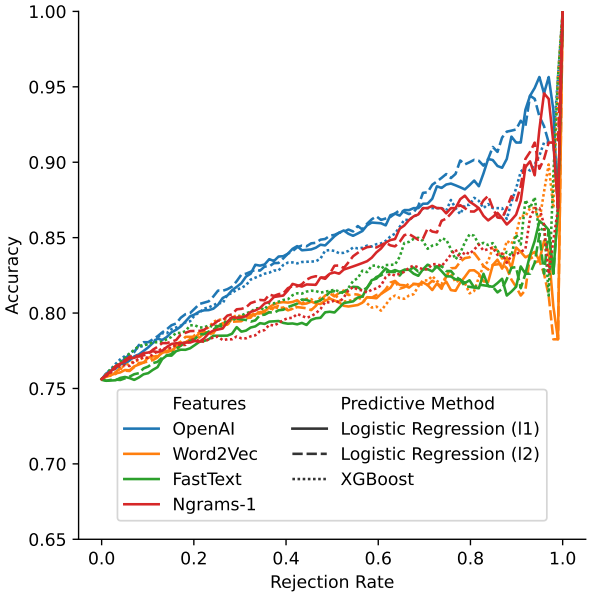

Figure 6: A comparison of the ARC curves for the different assessors of the highest accuracy LLM ("OpenAI/GPT-4o-2024-08-06") in our dataset.

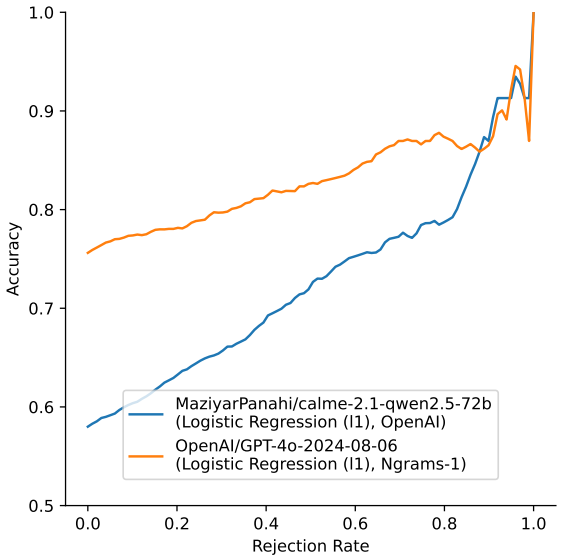

Figure 7: A comparison of two ARC curves in which one has a better performance at a PVR threshold of 0.8, and the other at a PVR threshold of 0.9.

also reported to capture predicability across the full range of error thresholds. Notice how, as expected, the best pairs all include LLMs in the upper quartile in terms of average accuracy (see Figure 10 in the Appendix). In particular, LLM-assessor pairs get a good score at a threshold of 0.8. This is to be expected when the LLMs are fairly good at the task (the best LLM has 75% accuracy), as the assessor can predict success most of the time. When the threshold is raised to 0.9 and above, we see a drastic drop in PVR, as this poses a greater requirement for assessors to make predictions that the LLM will fail. It is interesting, however, how the pair ranking is not preserved across the different thresholds.

To demonstrate how the choice of assessor impacts the ARC, in Figure 6 we compare the ARCs obtained with different assessors for the LLM "OpenAI/GPT-4o-2024-08-06", which achieves the highest accuracy on MMLU-Pro. The ARC varies substantially depending on assessor. In Figure 7, we examine two selected examples from our baselines. These show a case in which one of the two LLM-assessor pairs has a better PVR at a threshold of 0.8, and the other at a threshold of 0.9.

5.2 Out-of-Distribution Evaluation

To assess the robustness of our LLM-assessor pairs, we evaluated them on BBH after training on the train split of MMLU-Pro. Figure 8 replicates Figure 5 for the BBH benchmark. We use the same selection criteria as for the in-distribution results: the union of the top 5 PVR performers at each threshold. The lower values in Figure 8 highlight the difficulty of predicting performance OOD. Additionally, the top pairs differ from the in-distribution ones, suggesting the latter may not have the highest generalisation power. Other metrics for this OOD scenario are available in App. D.

6 Conclusion

`PredictaBoard` introduces a novel benchmark concept, evaluating pairs of models and validity

age researchers to design AI systems and assessors capable of operating under minimal error tolerance.

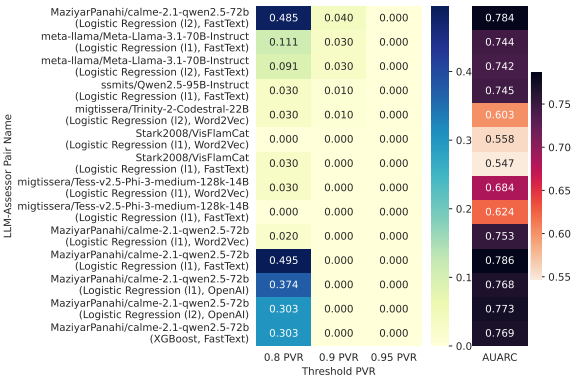

Figure 8: PVR at thresholds of 0.8, 0.9, and 0.95, and the area under the ARC curve for the union of the top 5 top LLM-assessor pairs at each threshold (OOD).

predictors (*assessors*). This aligns with the notion of validity predictability (Zhou et al., 2024a), highlighting how uncertainty on errors or safety, initially perceived as aleatoric, can become epistemic through pattern discovery (Hüllermeier and Waegeman, 2021). Leveraging external predictors to extract instance-level patterns contributes to making AI systems predictable and understandable, something that intrinsic uncertainty estimation falls short of. This stresses the critical importance of joint progress on LLMs and their assessors. An LLM is only as predictable as the quality of its assessors, and an assessor method is only effective if it performs well for state-of-the-art LLMs. `PredictaBoard` creates a unique opportunity to explore advancements in both LLMs and assessors, offering potential gains on these two fronts.

This paper aimed to release an initial version of the benchmark and allow the community to guide its future extensions, that can possibly include developing assessors that function across multiple LLMs and predicting safety indicators (Zhang et al., 2024; Mazeika et al., 2024), rather than performance. Additionally, comparing with human performance as assessors, as examined in recent studies (Carlini, 2024; Vafa et al., 2024b; Gao et al., 2024; Zhou et al., 2024b), could provide insights into the differences in LLM predictability from automated and human perspectives.

7 Limitations

There are some limitations in the current version of `PredictaBoard`. First, our baselines only include external anticipative assessors. In principle, `PredictaBoard` allows for non-external assessors (including self-confidence and latent LLM's embeddings, as discussed in Sec. 3) and can be easily extended to consider the output of the model as well, using "verifiers" instead of assessors (although not relying on outputs has some advantages, highlighted in Sec. 2). Additional conditions could also be considered, such as the invertibility of the assessor—-whether, by using the assessor, one can generate inputs that ensure the model's performance exceeds a given score, given constraints on the input. This capability would be particularly valuable for red-teaming applications or enhancing explainability (e.g., generating counterfactuals). Considering variations for these additional conditions and others (like the computational cost of the LLM-assessor pair) could allow to study properties of the assessors, such as the exploration of scaling laws for pairs of LLMs and assessors, or explore situations where the assessor cannot be n times more costly than the LLM. We leave all these considerations for future versions and specific competitions.

Some obvious limitations are the number of metrics, datasets and baseline methods. For metrics, one could also consider Pareto-dominance rather than single metrics, plotting the evolution of LLMs and assessors bidimensionally. Similarly, we could have considered cost-based metrics, such as those discussed in Hendrickx et al., 2024, Section 3.3, assigning a relative cost to rejections with respect to errors and computing the total cost using a fixed rejection threshold. Unlike the non-rejection rate metric we use (which is a specific case of cost-based metrics), these metrics require the definition of application-specific rejection costs and a maximum total cost, making them more complex to use in a standardised benchmark. For datasets, it is not always easy to find good sources covering a wide range of state-of-the-art LLMs at the instance level (Burnell et al., 2023), but more and more benchmarks with instance-level test results are available to be included, such as the remaining benchmarks involved in the Open-LLM Leaderboard, among others. These could be used to train assessors on more diverse data as well as for evaluating them out of distribution. In future releases, we thus plan to expand the coverage of our analysis and the datasets included in `PredictaBoard`.

Acknowledgments

Lorenzo Pacchiardi received funding from US DARPA (grant HR00112120007, RECoG-AI) and Open Philanthropy, and computational support from OpenAI through the Researcher Access Program. Ben Slater received funding from an ESRC scholarship (ES/P000738/1). Konstantinos Voudouris received funding from the Templeton World Charity Foundation grant (Major Transitions in the Evolution of Cognition: TWCF-2020-20539) and was supported by Helmholtz Zentrum München Deutsches Forschungszentrum für Gesundheit und Umwelt (GmbH). Jose Hernandez-Orallo and Fernando Martínez-Plumed received funding from CIPROM/2022/6 (FASSLOW) and IDIFEDER/2021/05 (CLUSTERIA) funded by Generalitat Valenciana, the EC H2020-EU grant agreement No. 952215 (TAILOR), and Spanish grant PID2021-122830OB-C42 (SFERA) funded by MCIN/AEI/10.13039/501100011033 and "ERDF A way of making Europe" CÁTEDRA ENIA-UPV en IA DESARROLLO SOSTENIBLE, TSI-100930-2023-9. This research project has benefitted from the Microsoft Accelerate Foundation Models Research (AFMR) grant program. In compliance with the recommendations of Burnell et al. (2023) on the reporting of evaluation results in AI, we include all the results at the instance level (https://github.com/Kinds-of-Intelligence-CFI/PredictaBoard).

References

Saleema Amershi, Max Chickering, Steven M Drucker, Bongshin Lee, Patrice Simard, and Jina Suh. 2015. Modeltracker: Redesigning performance analysis tools for machine learning. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*, pages 337–346.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. 2016. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*.

Usman Anwar, Abulhair Saparov, Javier Rando, Daniel Paleka, Miles Turpin, et al. 2024. Foundational challenges in assuring alignment and safety of large language models. *Preprint*, arXiv:2404.09932.

Terje Aven. 2016. Risk assessment and risk management: Review of recent advances on their foundation. *European journal of operational research*, 253(1):1–13.

Amos Azaria and Tom Mitchell. 2023. The internal state of an LLM knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 967–976, Singapore. Association for Computational Linguistics.

Nicholas J Bahr. 2014. *System safety engineering and risk assessment: a practical approach*. CRC press.

Gagan Bansal, Jennifer Wortman Vaughan, Saleema Amershi, Eric Horvitz, Adam Fourney, Hussein Mozannar, Victor Dibia, and Daniel S Weld. 2024. Challenges in human-agent communication. Accessed: 2024-12-10.

Yoshua Bengio, Michael K. Cohen, Nikolay Malkin, Matt MacDermott, Damiano Fornasiere, Pietro Greiner, and Younesse Kaddar. 2024. Can a Bayesian oracle prevent harm from an agent? *Preprint*, arXiv:2408.05284.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146.

Andrew P Bradley. 1997. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159.

Lennart Burger, Fred A Hamprecht, and Boaz Nadler. 2024. Truth is universal: Robust detection of lies in llms. *Advances in Neural Information Processing Systems*, 37:138393–138431.

Ryan Burnell, Wout Schellaert, John Burden, Tomer D Ullman, Fernando Martinez-Plumed, Joshua B Tenenbaum, Danaja Rutar, Lucy G Cheke, Jascha Sohl-Dickstein, Melanie Mitchell, Douwe Kiela, Murray Shanahan, Ellen M. Voorhees, Anthony G. Cohn, Joel Z Leibo, and Jose Hernandez-Orallo. 2023. Rethink reporting of evaluation results in AI. *Science*, 380(6641):136–138.

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, et al. 2023. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. *arXiv preprint arXiv:2312.09390*.

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

Nicholas Carlini. 2024. A GPT-4 capability forecasting challenge. https://nicholas.carlini.com/writing/llm-forecast/question/Capital-of-Paris. Accessed: 2024-09-08.

Paul Francis Christiano, Buck Shlegeris, and Dario Amodei. 2018. Supervising strong learners by amplifying weak experts. *ArXiv*, abs/1810.08575.

David "davidad" Dalrymple, Joar Skalse, Yoshua Bengio, Stuart Russell, Max Tegmark, Sanjit Seshia, Steve Omohundro, Christian Szegedy, Ben Goldhaber, Nora Ammann, Alessandro Abate, Joe Halpern, Clark Barrett, Ding Zhao, Tan Zhi-Xuan, Jeannette Wing, and Joshua Tenenbaum. 2024. Towards guaranteed safe AI: A framework for ensuring robust and reliable AI systems. *Preprint*, arXiv:2405.06624.

DARPA. 2024. Artificial Intelligence Quantified (AIQ). https://sam.gov/opp/78b028e5fc8b4953acb74fabf712652d/view. Accessed: 22nd October, 2024.

Kaustubh D. Dhole, Varun Gangal, Sebastian Gehrmann, Aadesh Gupta, Zhenhao Li, et al. 2022. Nl-augmenter: A framework for task-sensitive natural language augmentation. *Preprint*, arXiv:2112.02721.

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Ruhle, Laks VS Lakshmanan, and Ahmed Hassan Awadallah. 2024. Hybrid llm: Cost-efficient and quality-aware query routing. *arXiv preprint arXiv:2404.14618*.

Patricia Drapal, Ricardo B.C. Prudêncio, and Telmo M. Silva Filho. 2024a. Towards explainable evaluation: Explaining predicted performance using local performance regions. *Applied Soft Computing*, 167:112351.

Patricia Drapal, Telmo Silva-Filho, and Ricardo B. C. Prudêncio. 2024b. Meta-Learning and Novelty Detection for Machine Learning with Reject Option. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

Susan E Embretson and Steven P Reise. 2013. *Item response theory*. Psychology Press.

Farshid Faal, Ketra Schmitt, and Jia Yuan Yu. 2023. Reward modeling for mitigating toxicity in transformer-based language models. *Applied Intelligence*, 53(7):8421–8435.

Javier Ferrando, Oscar Balcells Obeso, Senthooran Rajamanoharan, and Neel Nanda. 2025. Do i know this entity? knowledge awareness and hallucinations in language models. In *The Thirteenth International Conference on Learning Representations*.

César Ferri, Peter Flach, and José Hernández-Orallo. 2004. Delegating classifiers. In *Proceedings of the twenty-first international conference on Machine learning*, page 37.

Yuan Gao, Dokyun Lee, Gordon Burtch, and Sina Fazelpour. 2024. Take caution in using llms as human surrogates: Scylla ex machina. *arXiv preprint arXiv:2410.19599*.

Tilmann Gneiting and Adrian E Raftery. 2007. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378.

Nicholas Goldowsky-Dill, Bilal Chughtai, Stefan Heimersheim, and Marius Hobbhahn. 2025. Detecting strategic deception using linear probes. *arXiv preprint arXiv:2502.03407*.

Hacking Policy Council. 2023. AI red teaming-legal clarity and protections needed.

Kilian Hendrickx, Lorenzo Perini, Dries Van der Plas, Wannes Meert, and Jesse Davis. 2024. Machine learning with a reject option: A survey. *Machine Learning*, 113(5):3073–3110.

Dan Hendrycks and Kevin Gimpel. 2016. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*.

Dan Hendrycks, Mantas Mazeika, and Thomas Woodside. 2023. An overview of catastrophic ai risks. *Preprint*, arXiv:2306.12001.

Danny Hernandez and Tom B Brown. 2020. Measuring the algorithmic efficiency of neural networks. *arXiv preprint arXiv:2005.04305*.

José Hernández-Orallo, Wout Schellaert, and Fernando Martínez-Plumed. 2022. Training on the test set: Mapping the system-problem space in AI. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 12256–12261.

Qitian Jason Hu, Jacob Bieker, Xiuyu Li, Nan Jiang, Benjamin Keigwin, Gaurav Ranganath, Kurt Keutzer, and Shriyash Kaustubh Upadhyay. 2024. Routerbench: A benchmark for multi-llm routing system. *arXiv preprint arXiv:2403.12031*.

Qian Huang, Jian Vora, Percy Liang, and Jure Leskovec. 2024. Benchmarking large language models as AI research agents.

Eyke Hüllermeier and Willem Waegeman. 2021. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine learning*, 110(3):457–506.

Andrew Ilyas, Sung Min Park, Logan Engstrom, Guillaume Leclerc, and Aleksander Madry. 2022. Datamodels: Predicting predictions from training data. *arXiv preprint arXiv:2202.00622*.

Shumaila Javaid, Nasir Saeed, and Bin He. 2024. Large language models for uavs: Current state and pathways to the future. *Preprint*, arXiv:2405.01745.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Sanyam Kapoor, Nate Gruver, Manley Roberts, Katherine M. Collins, Arka Pal, Umang Bhatt, Adrian Weller, Samuel Dooley, Micah Goldblum, and Andrew Gordon Wilson. 2024. Large language models must be taught to know what they don't know. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Zachary Kenton, Noah Y Siegel, János Kramár, Jonah Brown-Cohen, Samuel Albanie, Jannis Bulian, Rishabh Agarwal, David Lindner, Yunhao Tang, Noah D Goodman, et al. 2024. On scalable oversight with weak LLMs judging strong LLMs. *arXiv preprint arXiv:2407.04622*.

Geunwoo Kim, Pierre Baldi, and Stephen McAleer. 2024. Language models can solve computer tasks. *Advances in Neural Information Processing Systems*, 36.

A. Kipnis, K. Voudouris, L. M. Schulze Buschoff, and E. Schulz. 2024. metabench: A sparse benchmark to measure general ability in large language models. *arXiv preprint arXiv:2407.12844*.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *International conference on machine learning*, pages 957–966. PMLR.

John P Lalor, Hao Wu, and Hong Yu. 2016. Building an evaluation scale using item response theory. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, volume 2016, page 648. NIH Public Access.

Chia-Hsuan Lee, Hao Cheng, and Mari Ostendorf. 2023. Orchestrallm: Efficient orchestration of language models for dialogue state tracking. *arXiv preprint arXiv:2311.09758*.

Jan Leike, David Krueger, Tom Everitt, Miljan Martic, Vishal Maini, and Shane Legg. 2018. Scalable agent alignment via reward modeling: a research direction. *ArXiv*, abs/1811.07871.

Nancy G Leveson. 2002. System safety engineering: Back to the future. *Massachusetts Institute of Technology*.

Nancy G Leveson. 2016. *Engineering a safer world: Systems thinking applied to safety*. The MIT Press.

Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. 2017. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Teaching models to express their uncertainty in words. *Transactions on Machine Learning Research*.

Keming Lu, Hongyi Yuan, Runji Lin, Junyang Lin, Zheng Yuan, Chang Zhou, and Jingren Zhou. 2023. Routing to the expert: Efficient reward-guided ensemble of large language models. *arXiv preprint arXiv:2311.08692*.

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

Fernando Martínez-Plumed, Ricardo BC Prudêncio, Adolfo Martínez-Usó, and José Hernández-Orallo. 2019. Item response theory in ai: Analysing machine learning classifiers at the instance level. *Artificial intelligence*, 271:18–42.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David Forsyth, and Dan Hendrycks. 2024. HarmBench: A standardized evaluation framework for automated red teaming and robust refusal.

Tomas Mikolov. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 3781.

Malik Sajjad Ahmed Nadeem, Jean-Daniel Zucker, and Blaise Hanczar. 2009. Accuracy-rejection curves (arcs) for comparing classification methods with a reject option. In *Proceedings of the third International Workshop on Machine Learning in Systems Biology*, volume 8 of *Proceedings of Machine Learning Research*, pages 65–81, Ljubljana, Slovenia. PMLR.

Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas Tezak, Jong Wook Kim, Chris Hallacy, et al. 2022. Text and code embeddings by contrastive pretraining. *arXiv preprint arXiv:2201.10005*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

David Owen. 2024. How predictable is language model benchmark performance? *arXiv preprint arXiv:2401.04757*.

Lorenzo Pacchiardi, Lucy G. Cheke, and José Hernández-Orallo. 2024. 100 instances is all you need: predicting the success of a new LLM on unseen data by testing on a few instances. *Preprint*, arXiv:2409.03563.

Yudi Pawitan and Chris Holmes. 2024. Confidence in the reasoning of large language models. *arXiv preprint arXiv:2412.15296*.

F. M. Polo, L. Weber, L. Choshen, Y. Sun, G. Xu, and M. Yurochkin. 2024. tinybenchmarks: evaluating llms with fewer examples. *arXiv preprint arXiv:2402.14992*.

Yangjun Ruan, Chris J Maddison, and Tatsunori Hashimoto. 2024. Observational scaling laws and the predictability of language model performance. *arXiv preprint arXiv:2405.10938*.

Marija Šakota, Maxime Peyrard, and Robert West. 2024. Fly-swat or cannon? cost-effective language model choice via meta-modeling. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 606–615.

Wout Schellaert, Fernando Martínez-Plumed, and José Hernández-Orallo. 2024. Analysing the predictability of language model performance. *ACM Transactions on Intelligent Systems and Technology*.

Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. 2024. "do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models. *Preprint*, arXiv:2308.03825.

Tal Shnitzer, Anthony Ou, Mírian Silva, Kate Soule, Yuekai Sun, Justin Solomon, Neil Thompson, and Mikhail Yurochkin. 2023. Large Language Model Routing with Benchmark Datasets.

Ola Shorinwa, Zhiting Mei, Justin Lidard, Allen Z Ren, and Anirudha Majumdar. 2024. A survey on uncertainty quantification of large language models: Taxonomy, open research challenges, and future directions. *arXiv preprint arXiv:2412.05563*.

Grigori Sidorov, Francisco Velasquez, Efstathios Stamatatos, Alexander Gelbukh, and Liliana Chanona-Hernández. 2014. Syntactic n-grams as machine learning features for natural language processing. *Expert Systems with Applications*, 41(3):853–860.

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

Mark Steyvers, Heliodoro Tejeda, Aakriti Kumar, Catarina Belem, Sheer Karny, Xinyue Hu, Lukas W Mayer, and Padhraic Smyth. 2025. What large language models know and what people think they know. *Nature Machine Intelligence*, pages 1–11.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, , and Jason Wei. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher Manning. 2023. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5433–5442, Singapore. Association for Computational Linguistics.

Keyon Vafa, Ashesh Rambachan, and Sendhil Mullainathan. 2024a. Do large language models perform the way people expect? measuring the human generalization function. In *Forty-first International Conference on Machine Learning*.

Keyon Vafa, Ashesh Rambachan, and Sendhil Mullainathan. 2024b. Do large language models perform the way people expect? measuring the human generalization function. *arXiv preprint arXiv:2406.01382*.

Clara Vania, Phu Mon Htut, William Huang, Dhara Mungra, Richard Yuanzhe Pang, Jason Phang, Haokun Liu, Kyunghyun Cho, and Samuel R. Bowman. 2021. Comparing test sets with item response theory. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1141–1158, Online. Association for Computational Linguistics.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Preprint*, arXiv:2406.01574.

Jiaxin Wen, Ruiqi Zhong, Akbir Khan, Ethan Perez, Jacob Steinhardt, Minlie Huang, Samuel R. Bowman, He He, and Shi Feng. 2024. Language models learn to mislead humans via rlhf. *Preprint*, arXiv:2409.12822.

Marcus Williams, Micah Carroll, Adhyyan Narang, Constantin Weisser, Brendan Murphy, and Anca Dragan. 2024. Targeted manipulation and deception emerge when optimizing LLMs for user feedback. *arXiv preprint arXiv:2411.02306*.

Robert L Winkler. 1994. Evaluating probabilities: Asymmetric scoring rules. *Management Science*, 40(11):1395–1405.

Qinyuan Ye, Harvey Yiyun Fu, Xiang Ren, and Robin Jia. 2023. How predictable are large language model capabilities? a case study on BIG-bench. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Zhexin Zhang, Leqi Lei, Lindong Wu, Rui Sun, Yongkang Huang, Chong Long, Xiao Liu, Xuanyu Lei, Jie Tang, and Minlie Huang. 2024. SafetyBench: Evaluating the safety of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1:*

Long Papers), pages 15537–15553, Bangkok, Thailand. Association for Computational Linguistics.

Lexin Zhou, Fernando Martínez-Plumed, José Hernández-Orallo, Cèsar Ferri, and Wout Schellaert. 2022. Reject before you run: Small assessors anticipate big language models. In *EBeM@ IJCAI*.

Lexin Zhou, Pablo A. Moreno-Casares, Fernando Martínez-Plumed, John Burden, Ryan Burnell, Lucy Cheke, Cèsar Ferri, Alexandru Marcoci, Behzad Mehrbakhsh, Yael Moros-Daval, Seán Ó hÉigeartaigh, Danaja Rutar, Wout Schellaert, Konstantinos Voudouris, and José Hernández-Orallo. 2024a. Predictable artificial intelligence. *Preprint*, arXiv:2310.06167.

Lexin Zhou, Wout Schellaert, Fernando Martínez-Plumed, Yael Moros-Daval, Cèsar Ferri, and José Hernández-Orallo. 2024b. Larger and more instructable language models become less reliable. *Nature*, 634:61–68.

Lianghui Zhu, Xinggang Wang, and Xinlong Wang. 2023. JudgeLM: Fine-tuned large language models are scalable judges.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

A AI ecosystems, predictability and assessor models

In this appendix, we provide formal definitions of predictability as used in this paper, adapted from Zhou et al. (2024a). In this regard, we model the AI system and its interactions within an **AI ecosystem** (ranging from single AI systems interacting with individual users for specific tasks to complex socio-technical environments, with different levels of granularity) as follows: $\mathcal{I}$ is the set of problem instances (e.g., input prompts). $\mathcal{S}$ is the set of AI systems considered (e.g., LLMs). $\mathcal{U}$ is the set of users or operators interacting with the AI systems. $\mathcal{O}$ is the set of possible outputs from the AI systems. $\mathcal{R} \subseteq \mathcal{I} \times \mathcal{S} \times \mathcal{U} \times \mathcal{O}$ capture the relationships and interactions among instances, systems, users, and outputs. An AI ecosystem at time t is then represented as a tuple $\mathsf{E}_t = \langle \mathcal{I}_t, \mathcal{S}_t, \mathcal{U}_t, \mathcal{R}_t \rangle$ where the components may change over time.

We consider a distribution over ecosystems denoted as $\mathcal{E}_t$, and the complete history of interactions up to time t is represented by: $\mathsf{H}_{\leq t} = \langle \mathsf{E}_{\leq t}, \mathsf{O}_{<t}, V_{<t} \rangle$, where V_t is a random variable indicating the validity of outputs at time t (e.g., whether the LLM provides a correct answer for instance i).

Predictability is then formalised through the conditional probability distribution: $p(V_{t+h} \mid \mathsf{H}_{\leq t})$, which represents the probability of observing a valid output at a future time $t+h$, given the history up to time t. We define unpredictability $\mathbb{Q}$ as the minimum expected loss over a family of predictors $\mathcal{F}_b$ constrained by resource budgets b:

$$\mathbb{Q}(p, \mathcal{H}_t, \mathcal{F}_b) := \min_{\hat{p} \in \mathcal{F}_b} \mathop{\mathbb{E}}_{\substack{\mathsf{H}_{\leq t} \sim \mathcal{H}_{\leq t} \\ v \sim p(V_{t+h} | \mathsf{H}_t)}} S(\hat{p}(V_{t+h} | H_{\leq t}), v) \tag{1}$$

where S is a scoring function assessing the accuracy of predictions, such as the Brier Score.

In our context of `PredictaBoard`, an **assessor model** a belongs to a family of predictors $\mathcal{F}_b$, constrained by computational resources and the information it relies on (e.g., not using the LLM's outputs). An assessor is thus defined as $a : \mathcal{I} \to [0, 1]$, predicting the validity of the LLM's output on individual instances, namely $P(v_i = 1 \mid x_i))$, where x_i is the feature representation of instance i, and $v_i \in \{0, 1\}$ indicates the validity (e.g. success) of the LLM on that instance. The goal is thus to find an assessor that minimises unpredictability $\mathbb{Q}$ by minimising the expected loss: $\min_{a \in \mathcal{F}_b} \frac{1}{n} \sum_{i=1}^{n} S\big(a(x_i), v_i\big)$, over a test set of n instances, where S is the scoring function.

B Assessor metrics

B.1 Area Under the Receiving Operating Characteristic Curve (AUROC)

The AUROC assesses the quality of a binary probabilistic classifier by measuring its ability to discriminate between positive and negative instances across various thresholds. Specifically, the AUC plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at different threshold levels on this probability, obtaining a curve known as the Receiver Operating Characteristic Curve (ROC) curve. The AUROC is then calculated as the integral of this curve, providing a single scalar value that summarises the overall performance of the considered binary classifier. An AUROC of 1 indicates perfect discrimination, while an AUROC of 0.5 suggests no better performance than random chance. These extreme values are insensitive to the ratio of positively and negatively labelled instances in the dataset; thus, the AUROC can be seamlessly used to compare different scenarios where those ratios differ. This characteristic is particularly useful when comparing (LLM, assessor) pairs. However, the AUROC is insensitive to monotonic transformations of the probabilities predicted by the classifier. This implies that a miscalibrated classifier can still achieve a high AUROC. While increasing the AUROC will enhance the discrimination between the two classes, it does not necessarily improve the calibration of the classifier.

B.2 Brier Score

The Brier Score (BS) is equivalent to computing the mean squared error between the assessor predictions for each instance x_i and the actual success v_i:

$$\mathrm{BS} = \frac{1}{n} \sum_{i=1}^{n} \left(a(x_i) - v_i\right)^2 .$$

A perfect assessor would achieve a BS of 0, and larger scores indicate poorer predictions. The BS is an example of a strictly proper scoring rule (Gneiting and Raftery, 2007) — that is, a scoring method for probabilistic predictions that encourages recovery of the true data distribution when minimised. As such, the BS can be decomposed into calibration and refinement components (with the latter related to AUROC). This decomposition means that the BS

incentivises assessors to improve both calibration and discrimination.

However, the scale of the BS depends on the ratio of positive to negative elements v_i for the considered subject. Specifically, an assessor that always predicts the proportion of positive samples q in the dataset achieves a BS of $q(1-q)$ as $n \to \infty$.

B.3 Winkler's score

Winkler (1994) presented a generic way to correct binary scoring rules so that the score achieved by assessors that always predict the average success rate for subjects with different success rates is the same, and so can be easily compared. This relies on transforming a symmetric score (namely, for which $S(p,1) = S(1-p,0)$) into a non-symmetric one. Applying this transformation to the Brier Score leads to (Gneiting and Raftery, 2007, Sec. 3.2):

$$\mathrm{WS} = \frac{1}{n}\sum_{i=1}^{n}\frac{\alpha_i}{\beta_i},$$

where

$$\begin{aligned}\alpha_i &= [(1-c)^2 - (1-a(x_i))^2]\mathbf{1}\{v_i = 1\} \\ &\quad + (c^2 - a(x_i)^2)\mathbf{1}\{v_i = 0\}, \\ \beta_i &= c^2\mathbf{1}\{a(x_i) \le c\} + (1-c)^2\mathbf{1}\{a(x_i) > c\},\end{aligned}$$

where c is the average accuracy of the considered LLM and $\mathbf{1}$ is the indicator function. The score for the assessor predicting the observed success rate is 0 while that for a perfect assessor is 1.

C LLMs

Our dataset consists of the instance-level performances of various LLMs on `MMLU-Pro` (Wang et al., 2024) and the BIG-Bench-Hard (BBH, Suzgun et al., 2022), for a total of 11383 and 5761 instances respectively. The results for a selection of 38 open-source LLMs for both MMLU-Pro and BBH were obtained from HuggingFace's *Open LLM Leaderboard v2*, which ranks open-source LLMs on these benchmarks; the selection includes a diversity of models from several well-known families, each with different architectures (LLama, GPT, Qwen, Mistral, etc.) and parameter sizes (up to 95B parameters). Further, the results for 3 versions of GPT-4o for MMLU-Pro were obtained from the original repository (Wang et al., 2024). Table 2 includes the full list of considered LLMs.

Moreover, Figures 10 and 11 show the performance of the different LLMs on the `MMLU-Pro` and BBH benchmarks respectively.

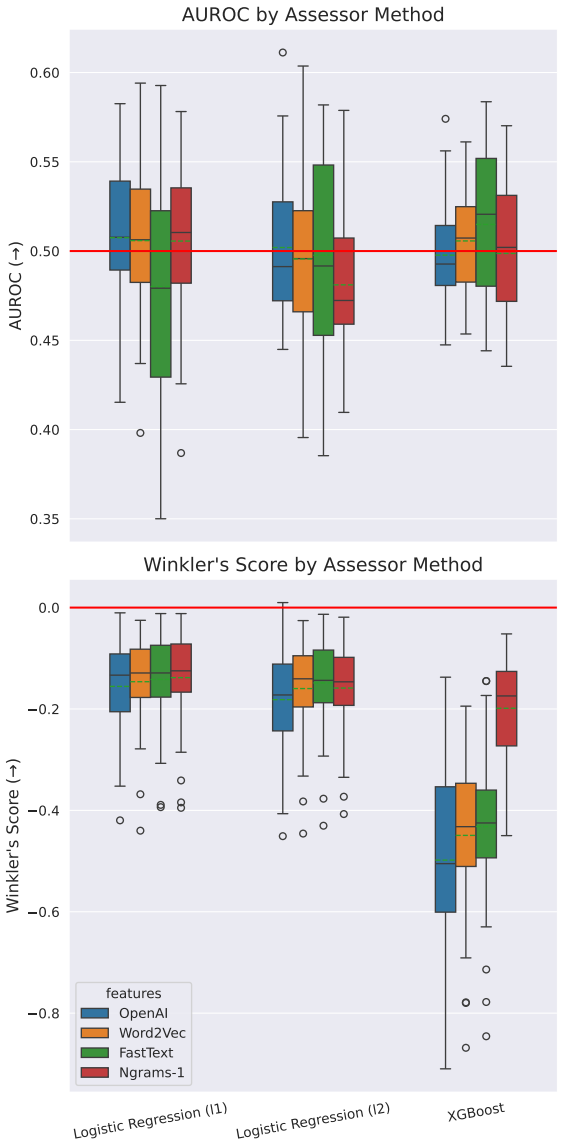

Figure 9: Distribution (over LLMs) of assessor-only performance metrics across different embedding schemes and classifiers, evaluated OOD on the BBH dataset.

D OOD AUROC and Winkler's score

In this section we replicate the results of Figure 4 for OOD assessment on the BBH dataset (see Figure 9). The results are worse, as expected, especially in calibration. This simply encourages more work on further abstraction and different partitions when training the assessors, and `PredictaBoard` is a tool for that. Note that OOD is a situation where simply extrapolating the accuracy of one benchmark to another is useless, and small improvements in OOD results can make a difference.

Table 2: List of LLMs used in our experimental setting. `PredictaBoard` includes instance-level results for MMLU-Pro and BBH for all of them except the three GPT-4o versions, for which BBH results are unavailable.

Index	HuggingFace's Model Name	Family	Size	Version
1	calme-2.1-qwen2.5-72b	Qwen	72B	2.5-Instruct
2	DCLM-7B	DCLM	7B	
3	gemma-2-2b-ORPO-jpn-it-abliterated-18-gemma-2-2b	Gemma	2B	v2 ORPO
4	GPT-4o-2024-05-13	GPT	-	4o-2024-05-13
5	GPT-4o-2024-08-06	GPT	-	4o-2024-08-06
6	GPT-4o-mini	GPT	-	4o-mini
7	GutenLaserPi	Mistral	7B	-
8	Hermes-3-Llama-3.1-70B	Llama	70B	3.1
9	Jallabi-34B	Llama	34B	v1.6
10	L3.1-ClaudeMaid-4x8B	Llama	4x8B	ClaudeMaid v1.0
11	LayleleFlamPi	Mistral	7B	-
12	leniachat-gemma-2b-v0	Gemma	2B	leniachat v0
13	leniachat-qwen2-1.5B-v0	Qwen	1.5B	leniachat v0
14	llama-3.1-8B-Galore-openassistant-guanaco	Llama	8B	3.1-Galore-openassistant-guanaco
15	Llama-3.1-8B-Lexi-Uncensored	Llama	8B	3.1-Lexi-Uncensored
16	Llama-3.1-8B-Lexi-Uncensored-V2	Llama	8B	3.1-Lexi-Uncensored-V2
17	Llama-3.1-8B-MagPie-Ultra	Llama	8B	3.1-MagPie-Ultra
18	Llama-3-70B-Synthia-v3.5	Llama	70B	Synthia-v3.5
19	Llama-3-8B-Synthia-v3.5	Llama	8B	Synthia-v3.5
20	llama-3-luminous-merged	Llama	8B	luminous-merged
21	Meta-Llama-3.1-70B-Instruct	Llama	70B	v3.1-Instruct
22	Mistral-7B-v0.1-signtensors-1-over-4	Mistral	7B	v0.1
23	Mistral-7B-v0.1-signtensors-3-over-8	Mistral	7B	v0.1
24	Mistral-7B-v0.1-signtensors-5-over-16	Mistral	7B	v0.1
25	Mistral-7B-v0.1-signtensors-7-over-16	Mistral	7B	v0.1
26	neural-chat-7b-v3-1	Neural Chat	7B	v3-1
27	neural-chat-7b-v3-2	Neural Chat	7B	v3-2
28	neural-chat-7b-v3-3	Neural Chat	7B	v3-3
29	pythia-410m-roberta-lr_8e7-kl_01-steps_12000-rlhf-model	Pythia	410M	-
30	Qwen2.5-95B-Instruct	Qwen	95B	2.5-Instruct
31	qwent-7b	Qwen	7b	-
32	solar-pro-preview-instruct	Solar Pro	22B	instruct
33	SuperHeart	Llama	8B	v3.1 SuperNova-Lite
34	Tess-3-7B-SFT	Mistral	7B	3-SFT
35	Tess-3-Mistral-Nemo-12B	Mistral	12B	Tess v3
36	Tess-v2.5.2-Qwen2-72B	Qwen	72B	Tess v2.5.2
37	Tess-v2.5-Phi-3-medium-128k-14B	Phi	14B	Tess v2.5.2.5
38	Trinity-2-Codestral-22B	Mistral	22B	Codestral v0.1
39	Trinity-2-Codestral-22B-v0.2	Mistral	22B	Codestral v0.1
40	VisFlamCat	Mistral	7B	-
41	Yarn-Llama-2-13b-128k	Llama	13B	v2

E Failure analysis

Models that always fail or always succeed are highly predictable. This is why we use the AUROC and Winkler's score as assessor metrics, because they are balanced, useful to counteract this effect and compare assessors for models of different accuracy. However, can we still find that more performant models are more predictable, even after controlling for this?

We explore this question in Figure 12, showing the relation between the accuracy and AUROC for all models and assessors respectively using the MMLU-Pro dataset (trained on the train split and evaluated on the test split). Recall that a classifier randomly guessing would produce AUROC of 0.5, while AUROC of 1 corresponds to perfect discrimination. We see a positive correlation, but the oriented interpretation is more interesting: assessors with high AUROC always correspond with models of high accuracy (the opposite is less clear). There are also two clear clusters in the plot, and the one on the top right has negative correlation. Moreover, in that cluster, the assessors using the most powerful features (the OpenAI embeddings) perform better.

Figure 13 replaces the AUROC with Winkler's score and shows a similar, but less clear behaviour, where again, higher Winkler's score implies higher LLM accuracy. Recall that, for the Winkler's score, a value of 0 corresponds to a constant assessor that always outputs the average accuracy; lower is worse and Winkler's score is 1 for perfect predictions. From this graph, two considerations can be made:

1. For the models with low accuracy, assessors are unable to get above the 0 (baseline level for a constant assessor) in terms of Winkler's score. Moreover, considering individually each LLM with low accruacy, the least powerful features (Ngrams-1) always lead to higher Winkler's score for the assessors, while the most powerful ones lead to much lower score

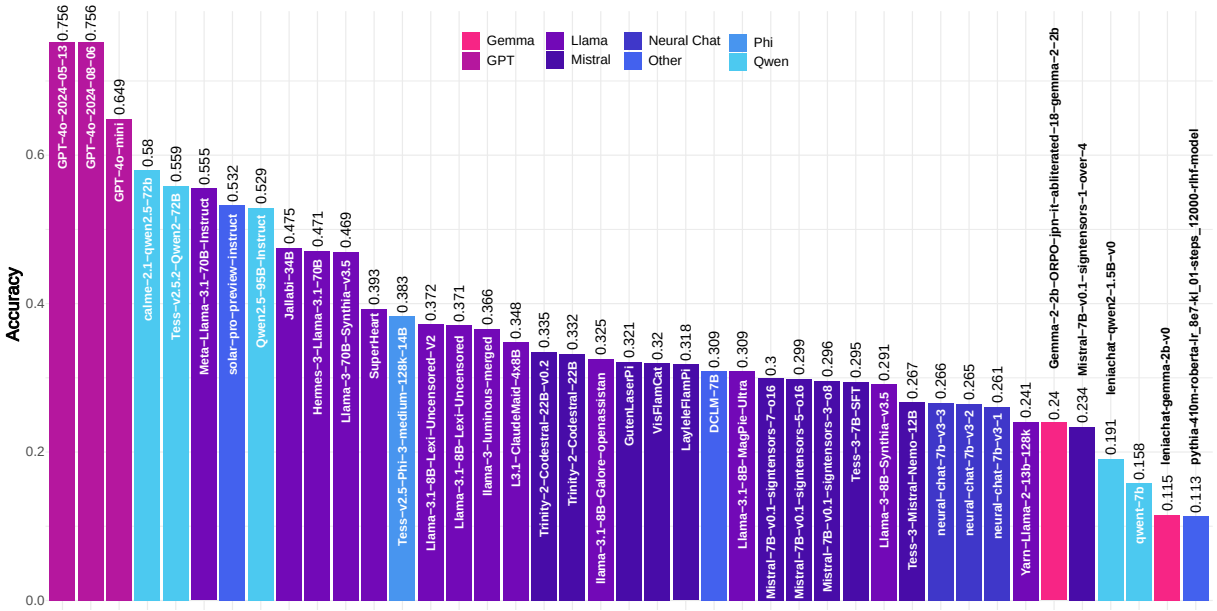

Figure 10: The performance of the LLMs in the MMLU-Pro dataset, expressed as the proportion of questions answered correct.

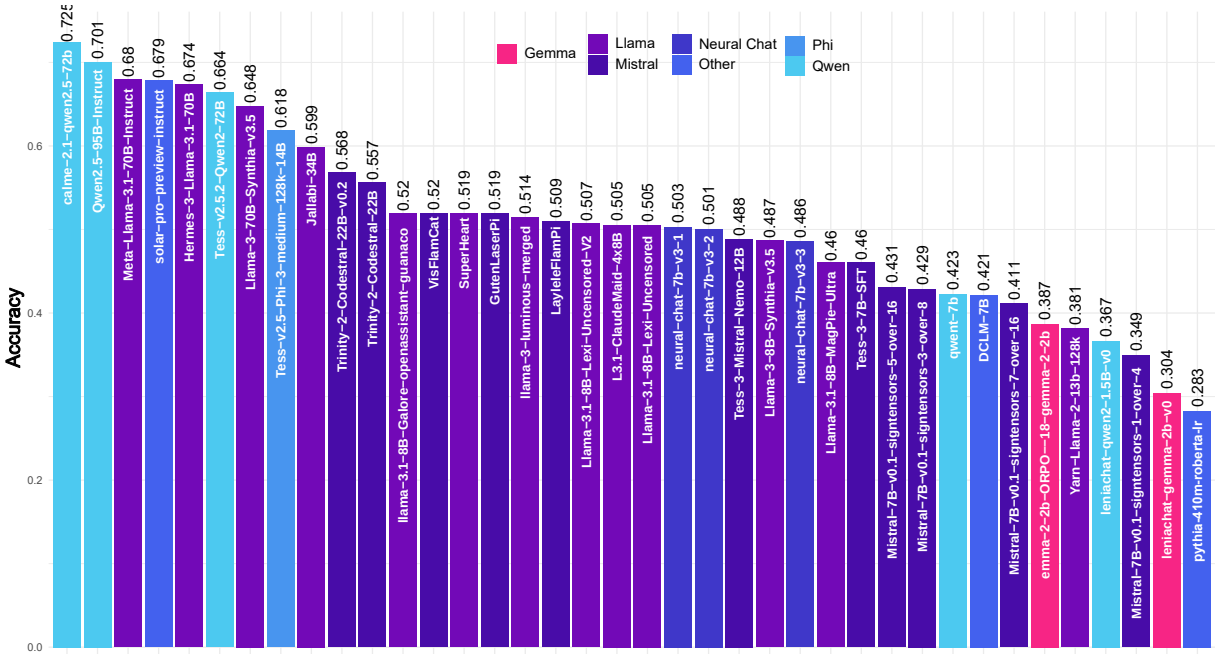

Figure 11: The performance of the LLMs in the BBH dataset, expressed as the proportion of questions answered correct.

(there is instead a mix for the two intermediate features). This suggests that most powerful features lead to the assessors overfitting on the training set, hence suggesting that these models have intrinsically poorly predictably.

2. Instead, for the models with high accuracy, the opposite pattern can be observed: the most powerful features (the OpenAI embeddings) lead to higher Winkler's score, indicating that these features are encoding a general pattern impacting LLM performance.

That influence suggests that we could inspect specific examples to see extreme differences between the actual and the predicted outcome. Indeed, assessors can be a very useful tool to analyse errors, such as using counterfactuals as in explainable AI (e.g., would the model fail if we changed the prompt in some particular way?). Also we can

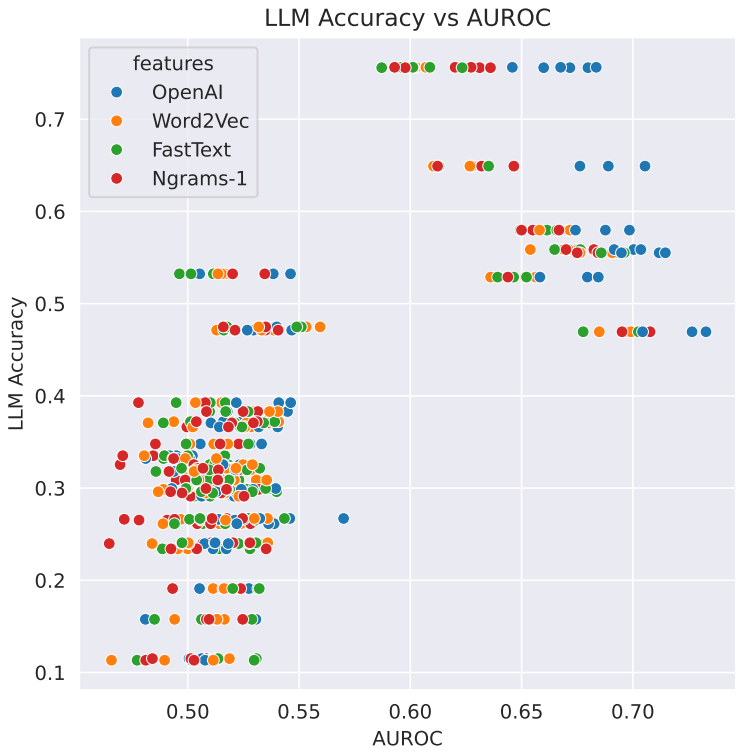

Figure 12: Relation between AUROC per assessor and accuracy per model on the test split of the MMLU-Pro dataset, for assessoras trained on the train split.

compare the confidence given by the assessor to different failures or successes. For instance, we can easily conduct a ranking of examples per benchmark according to their score predictability over different models and assessors, and how this relates to their difficulty (percentage of models that fail on the example). Theoretically, this should be related to the variance of a Bernoulli distribution, but it can be very insightful to explore deviations from this: failures of the model with high-confidence of success from the assessor and successes of the model with high-confidence of failure from the assessor.

To observe this, we selected the best LLM-assessor pair based on 0.8 PVR (`OpenAI/GPT-4o-2024-08-06` with Logistic Regression (l2) assessor based on OpenAI embeddings, see Fig. 5), and printed the lowest 5 instances in terms of assessor confidence which the LLM got right (Figure 14), and the highest 5 which the LLM got wrong (Figure 15). The high-confidence-but-wrong instances involve short questions, many of which seem straightforward, from several disciplines. In contrast, the low-confidence-but-correct instances involve very long question and many of them are law- or engineering-related. This shows how these failures, at least for this model, were ultimately caused by the base model and not by the assessor giving unreasonable estimates.

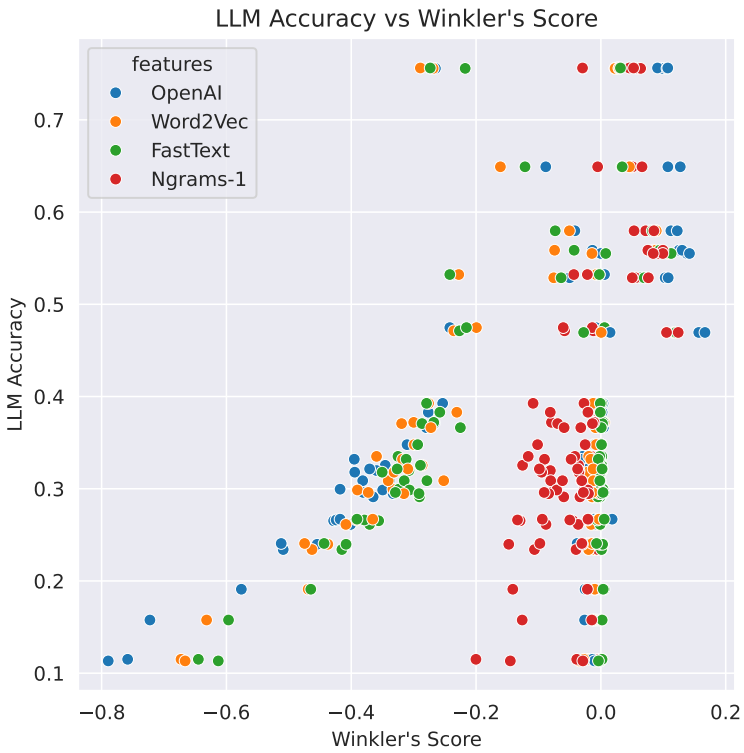

Figure 13: Relation between Winkler's score per assessor and accuracy per model on the test split of the MMLU-Pro dataset, for assessoras trained on the train split.

```
PROMPT 1400, Prediction: 0.35163472465364426:
Federal law prohibits "willingly and knowingly" taking cash in excess of $10,000 from the U.S. into a foreign country without first reporting the
↪  transaction in detail. An owner of a Detroit drug store takes his gross cash receipts each week into a city in Canada where he lives and does his
↪  banking. The office of the Deputy Atty. General learned that the owner was doing this, and indicted him on 10 counts of "willingly and knowingly"
↪  taking cash over $10,000 into a foreign country without reporting it. The owner's main defense is that he did not know of the law or that he was
↪  breaking it. The trial judge instructed the jury that mistake of law is no defense. He was convicted and appealed. Will the federal appellate court
↪  likely reverse the conviction?
A. No, because the owner's habitual actions imply intent to avoid reporting the cash.
B. No, the practice is so dangerous to the public interest that knowledge and specific intent are not required.
C. Yes, because willfulness clause requires proof of both knowledge of the law and a specific intent to commit the crime.
D. No, willfulness and knowledge are inferred by the habitual practice of transporting the cash.
E. Yes, because the owner was not intentionally breaking the law, he was simply unaware of it.
F. No, because ignorance of the law is not a valid defense.
G. Yes, because the owner is not a resident of the U.S. and therefore not subject to its laws.
H. No, because the owner is a business operator and therefore should be aware of such laws.
I. Yes, because treaties with Canada make all such reporting laws unenforceable.
J. Yes, because the owner was not given a fair chance to defend himself in court.
Answer:

PROMPT 1758, Prediction: 0.3642458853764882:
A man who owned a business believed that one of his employees was stealing computer equipment from the business. He decided to break into the employee's
↪  house one night, when he knew that the employee and her family would be away, to try to find and retrieve the equipment. The man had brought a
↪  picklock to open the employee's back door, but when he tried the door, he found that it was unlocked, so he entered. As the man was looking around
↪  the house, he heard sounds outside and became afraid. He left the house but was arrested by police on neighborhood patrol. What is the man's
↪  strongest defense to a burglary charge?
A. The back door to the house was unlocked.
B. The man was scared and left the house before committing a crime.
C. The man did not actually use the picklock.
D. The man was arrested outside, not inside, the house.
E. The man was only trying to retrieve his own property.
F. The man did not intend to commit a crime inside the house.
G. The man believed the stolen property was his.
H. The house was not occupied at the time of his entry.
I. The man did not take anything from the house.
Answer:

PROMPT 996, Prediction: 0.36962361800041293:
A wife is the beneficiary of a policy issued by an insurance company, insuring the life of her husband, now deceased. The policy contained a clause
↪  providing that double indemnity is payable in the event that death of the insured "results directly, and independently of all other causes, from
↪  bodily injury effected solely through external violent and unexpected means. "The husband was found dead in the chicken shed of his farm. His death
↪  resulted from wounds caused by a shotgun blast. The wife filed the necessary papers with the insurance company concerning proof of her husband's
↪  death. The insurance company admitted liability for the face amount of the policy but rejected the wife's claim for double indemnity. The wife then
↪  instituted suit against the insurance company demanding judgment according to the double indemnity provisions of the husband's insurance policy. At
↪  trial, the wife was called to testify about the events on the day of her husband's death. The wife said that she was in the kitchen when she heard a
↪  gunshot in the shed. As she rushed out of the house, she saw their neighbor running from the shed. The neighbor is present in court. As a witness,
↪  the wife was
A. competent, because she can provide a first-hand account of the incident.
B. incompetent, because she was not an eyewitness to the actual event.
C. incompetent, because her testimony is based on her perception of events.
D. competent, because she was present on the scene after the event occurred.
E. competent, because she had personal knowledge of the matter.
F. competent, because the neighbor is available to testify.
G. incompetent, because her testimony could potentially be biased.
H. incompetent, because she was testifying to facts occurring after her husband's death.
I. competent, because she can corroborate her account with the neighbor's testimony.
J. incompetent, because she had a personal interest in the outcome of the lawsuit.
Answer:

PROMPT 2162, Prediction: 0.38607054185111:
There is a state statute making it a misdemeanor "to falsely report a fire either intentionally or recklessly. " There were three college roommates who
↪  lived together in a small apartment. Two of the roommates decided to play a practical joke on the other roommate, which they liked to do from time to
↪  time because he was gullible. The two roommates were seated in the living room of their apartment. The other roommate was in an adjoining room and
↪  within earshot of the two roommates. Knowing that their roommate could hear their conversation, the two roommates falsely stated that a fire had been
↪  set at the student center at the college. After overhearing this conversation, the other roommate phoned the fire department and reported this
↪  information. Several fire trucks were dispatched to the college and determined the information to be false. If the two roommates are prosecuted for
↪  violating the aforementioned statute, they should be found
A. guilty, because they intentionally misled their roommate.
B. guilty, because they deliberately provided false information.
C. not guilty, because they did not directly call the fire department.
D. guilty, because they caused the false report to be made.
E. guilty, because they knowingly spread false information within earshot of their roommate.
F. guilty, because they are accomplices to their roommate.
G. not guilty, because it was a practical joke and they did not intend harm.
H. not guilty, because they didn't knowingly believe that their roommate would report the information to the fire department.
I. not guilty, because they didn't report the information to the fire department themselves.
J. not guilty, because they did not confirm the presence of a fire themselves.
Answer:

PROMPT 485, Prediction: 0.38794118768882585:
A defendant was on the first day of her new secretarial job when her boss called her into his office. The boss directly suggested that if the defendant
↪  did not go out on a date with him, she would be fired in one week. Every day during the remainder of the week, the boss approached the defendant with
↪  his demand, and the defendant refused to cooperate. At the end of the week, when the boss called the defendant into his office and again tried to
↪  pressure her to go out on a date with him, the defendant knocked him unconscious with a giant stapler and choked him to death. The defendant is tried
↪  for murder. In accordance with the following statute, the state relies at trial on the presumption of malice:"When the act of killing another is
↪  proved, malice aforethought shall be presumed, and the burden shall rest upon the party who committed the killing to show that malice did not exist.
↪  "If the defendant is convicted of first-degree murder and challenges her conviction on the grounds of the above statute, on appeal she will
A. lose, because the presumption may be rebutted.
B. win, because the statute violates due process.
C. lose, because the presumption of malice aforethought is constitutional.
D. win, because she acted in self-defense.
E. lose, because her actions were premeditated.
F. win, because the statute is unjust.
G. lose, because she did not show that malice did not exist.
H. win, because the statute is discriminatory.
I. lose, because she failed to overcome the presumption.
Answer:
```

Figure 14: Lowest 5 instances in terms of assessor confidence which the LLM got right. This is shown for the best LLM-assessor pair based on 0.8 PVR (`OpenAI/GPT-4o-2024-08-06` with Logistic Regression (l2) assessor based on OpenAI embeddings) and the MMLU-Pro dataset.

```
PROMPT 365, Prediction: 0.9677333347366117:
A store sells two items for $10 each. One item costs $5.25, while the other costs $6.50. What ratio of items at each price must be purchased in order to
↪  have an average markup based on the selling price of 40%?
A. 3 to 1
B. 4 to 3
C. 1 to 2
D. 3 to 2
E. 2 to 3
F. 2 to 5
G. 1 to 4
H. 1 to 3
I. 4 to 1
J. 5 to 3
Answer:

PROMPT 929, Prediction: 0.9578498958041702:
What is criterion related validity?
A. Criterion-related validity evaluates the test's ability to predict future or past performance.
B. Criterion-related validity is the testing of an individual's knowledge
C. Criterion-related validity measures the extent to which test scores are unaffected by external factors.
D. Criterion-related validity measures the test's consistency
E. Criterion-related validity assesses the degree to which a test captures a comprehensive range of abilities within a domain.
F. Criterion-related validity is the extent to which a test measures a theoretical construct or trait.
G. Criterion-related validity refers to the bias in a test's results
H. Criterion-related validity is the degree to which a test aligns with another widely accepted standard test.
I. Criterion-related validity is concerned with the extent to which a test correlates with a concurrent benchmark.
J. Criterion-related validity refers to the effectiveness of a test in predicting an individual's behavior in specified situations.
Answer:

PROMPT 1241, Prediction: 0.9559929993297027:
A store has 3 boxes of shirts. Each box has 4 packages with 7 shirts in each package. The expression 3 × (4 x 7) can be used to find the total number of
↪  shirts. Which expression can also be used to find the total number of shirts?
A. 14 × 3
B. 21 × 4
C. 12 × 7
D. 7 × 12
E. 14 × 4
F. 4 × 21
G. 12 × 3
H. 28 × 4
I. 28 × 7
J. 21 × 3
Answer:

PROMPT 902, Prediction: 0.9389530358257672:
Even though there is no such thing as a "typical cell" - for there are too many diverse kinds of cells - biologists have determined that there are two
↪  basic cell types. What are these two types of cells?
A. Single-celled and Multi-celled
B. Animal and Plant cells
C. Procaryotic and Eucaryotic
D. Diploid and Haploid cells
E. Photosynthetic and Non-photosynthetic cells
F. Vascular and Non-vascular cells
G. Prokaryotic and Eukaryotic
H. Somatic and Germ cells
I. Autotrophic and Heterotrophic cells
J. Aerobic and Anaerobic cells
Answer:

PROMPT 1229, Prediction: 0.9381125294629535:
Which of the following about meiosis is NOT true?
A. Meiosis produces two haploid gametes.
B. Homologous chromosomes join during synapsis.
C. Sister chromatids separate during meiosis I.
D. Crossing-over increases genetic variation in gametes.
Answer:
```

Figure 15: Highest 5 instances in terms of assessor confidence which the LLM got wrong. This is shown for the best LLM-assessor pair based on 0.8 PVR (`OpenAI/GPT-4o-2024-08-06` with Logistic Regression (l2) assessor based on OpenAI embeddings) and the MMLU-Pro dataset.