On the Evaluation of Capability Estimation Methods for Large Language Models

Qiang Hu[1*], **Jin Wen**[2 *], **Yao Zhang**[1†], **Maxime Cordy**[2], **Yongqiang Lyu**[1]

[1]School of Cyber Security, Tianjin University
[2]University of Luxembourg
qianghu0515@gmail.com, jin.wen@uni.lu, zzyy@tju.edu.cn, maxime.cordy@uni.lu, lyuyq@tju.edu.cn

Abstract

The emergence of large language models (LLMs) marks a transformative era in artificial intelligence (AI). However, systematically evaluating the capability of LLMs is challenging due to the necessity of a large number of labeled test data. To tackle this problem, in the conventional AI field, AutoEval has been proposed to estimate the capability of AI models without data labeling effort. Unfortunately, even though multiple AutoEval methods have been proposed, most are constructed for classification tasks and evaluated only on image datasets. As a result, their effectiveness for LLMs is unclear, as LLMs often target generation tasks. In this work, we introduce the first AutoEval benchmark specifically designed to estimate the capability of LLMs using unlabeled test data, AEBench. Besides existing AutoEval methods, AEBench also supports our designed method, which utilizes the correlation between data uncertainty and model ability for the capability estimation. In total, AEBench covers 12 AutoEval methods and 120 method combinations. Based on AEBench, we conducted a comprehensive study to explore the usefulness of AutoEval on LLMs. Experimental results on 10 datasets demonstrated that our designed uncertainty features-based methods perform the best in achieving the lowest estimation errors.

Code — https://github.com/lighteningbird/aebench

Introduction

Even though large language models (LLMs) have become increasingly popular across various applications due to their remarkable capabilities, measuring their capabilities is challenging due to the high cost of preparing labeled test data across different domains. Data preparation, especially annotation, is time-consuming and labor-intensive. Multiple works attempted to construct benchmark datasets with carefully labeled data to help evaluate LLMs, such as HumanEval (Chen et al. 2021) and EvalPlus (Liu et al. 2024). However, these datasets cover limited samples, for example, only 164 test samples are considered in HumanEval. As a result, the reported results cannot precisely reflect the capability of LLMs. How to comprehensively and efficiently evaluate LLMs is still an open problem.

In the conventional deep learning (DL) field, to efficiently evaluate DL models, AutoEval (Deng and Zheng 2021) framework has been introduced to estimate the ability of DL models using only unlabeled test data. Later on, multiple AutoEval methods (Deng et al. 2023) have been proposed with ever better estimation accuracy. Employing AutoEval for LLMs is a straightforward and promising direction to facilitate the efficient evaluation of LLMs. However, the effectiveness of AutoEval for LLMs is unclear due to the following three reasons, 1) some AutoEval methods (Hu et al. 2023) require training data to guide estimation, however, the pre-training data of LLMs are usually unknown; 2) most AutoEval methods (Deng et al. 2023) are constructed for classification tasks and evaluated on image datasets, but LLMs are generally used for generation tasks and datasets with different domains; and 3) some AutoEval methods (Saxena et al. 2024) are based on model ensembles that need to build multiple LLMs, which is not practical. In the end, to achieve efficient LLM evaluation, a benchmark to thoroughly support capability estimation methods is necessary.

To do so, in this paper, we construct the first benchmark, AEBench, for capability estimation for LLMs that supports different types of AutoEval methods. Based on their design construction, we categorize existing AutoEval methods into meta-model-free (MMF) and meta-model-based (MMB) two groups. MMF methods directly utilize the extracted features for the estimation, while MMB methods require meta-sets to construct the relationship between the extracted features and the model performance. Here, features indicate the information used for the estimation obtained from the input or model. According to the definition, we first collect methods in these two categories and then design a new series of MMB methods to boost the estimation performance. Concretely, we propose to learn the correlation between uncertainty information (uncertainty scores as features) and model performance, and then use this correlation for capability estimation. In total, AEBench covers 12 AutoEval methods, including five MMF methods (e.g., ATC (Garg et al. 2022)) and seven MMB methods (e.g., few-shot ICL accuracy estimation (Fu et al. 2023)). Besides, we consider two-level and multi-level feature combinations to enrich our benchmark and explore better AutoEval methods.

[*]These authors contributed equally.
[†]Corresponding author.

Based on AEBench, we conducted a comprehensive study to investigate the effectiveness of AutoEval methods for LLMs. Our study considers 10 datasets including both classification tasks (e.g., MMLU (Hendrycks et al. 2021)) and generation tasks (e.g., TriviaQA (Joshi et al. 2017)). In a more practical setting, we evaluate AutoEval methods on both in-distribution (Id) and out-of-distribution (OOD) scenarios. The experimental results demonstrated that our designed uncertainty-based MMB methods are more suitable for LLMs' capability estimation and Linear Regression is the most capable meta-model for MMB methods.

To summarize, the main contributions of this paper are: 1) We built the first benchmark, AEBench, to support AutoEval for LLMs. AEBench considers both meta-model-free and meta-model-based methods with a total of 12 AutoEval methods; 2) We conduct the first comprehensive study to explore the effectiveness of AutoEval methods for LLMs on 10 datasets; 3) We consider both ID and OOD testing scenarios for AutoEval for LLMs, which is crucial for real-world usage; 4) we demonstrated that uncertainty-based MMB methods perform the best over our benchmark.

Background and Related Work

Automatic Model Evaluation (AutoEval)

Automatic Model Evaluation (AutoEval) (Deng and Zheng 2021) for machine learning models is a problem to access the capability of models directly using unlabeled test data. In this direction, multiple methods have been proposed. Deng *et al.* designed a multi-task framework and utilized an auxiliary task to estimate the accuracy of classifiers under new environment (Deng, Gould, and Zheng 2021). Some works try to build a regression model the learn the relationship between the accuracy of models and the distance between seen and unseen data (Deng and Zheng 2021) for the accuracy estimation. Though such regression-based methods look promising to estimate model performance, it is still challenging due to its requirement of labeled OOD (unseen) data for building the regression, which may be costly on LLMs (Maggio, Bouvier, and Dreyfus-Schmidt 2022). To tackle the labeling problem, confidence based methods that only depend on model output to estimate the performance, i.e. ATC (Garg et al. 2022) and ATC assumes all test samples with confidence lower than a threshold are wrong, and verse vice. COT measures the distance between source and target distribution to estimate the test error, which is based on the optimal transport of model confidences. Different from existing works that mainly focus on classification tasks and conventional classifiers, we extend the AutoEval problem to LLMs and comprehensively study this problem.

Confidence Measurement

Confidence of neural networks refers to the probability of the prediction being correct (Niculescu-Mizil and Caruana 2005). Existing works demonstrated that the confidence value can be used as an indicator to estimate errors in inputs (Hendrycks and Gimpel 2017), thus estimating the performance of models. However, research revealed that there is an overconfident issue (Nguyen, Yosinski, and Clune 2015) in deep learning models and simply using confidence to estimate model capability could be misleading. To solve this problem, some works propose to calibrate such confidence values for reliability. Among different calibration methods, temperature scaling stands out as a simple but effective post-processing method without additional training effort (Guo et al. 2017). Meanwhile, negative log-likelihood is a widely used measurement to evaluate the quality of probabilistic models, which is calculated by cross-entropy loss of model output (Hastie, Friedman, and Tibshirani 2001). Correspondingly, negative log-likelihood can be extended into generation tasks to evaluate the confidence of large language models, where models typically generate output based on Transformer-like probabilistic auto-regressive models (Vaswani et al. 2017). Existing works most focus on item-level confidence measurement and calibration, while we synthesize different confidence based features into data set levels to estimate the performance of large language models.

Uncertainty Measurement

In contrast to confidence, model prediction with higher uncertainty is more likely to be inaccurate (Snoek et al. 2019). There exists plenty of work about quantifying the uncertainty of DL models, which can be mainly categorized into two types: aleatoric uncertainty and epistemic uncertainty (Der Kiureghian and Ditlevsen 2009). Aleatoric uncertainty, also known as statistical uncertainty, is the inherent noise in the data, and it can be estimated by the model output probability values (Bao et al. 2023). Epistemic uncertainty is mainly about model uncertainty from its inner shortness, and it can be mitigated by collecting more data or improving model architecture (Der Kiureghian and Ditlevsen 2009). In this paper, we focus on the first uncertainty type. Multiple relevant uncertainty metrics have been proposed such as Margin (Scheffer, Decomain, and Wrobel 2001) and DeepGini (Feng et al. 2020). Margin measures the probabilities difference between the most likely and second most likely classes. A small Margin score indicates the model is uncertain against the data. DeepGini is designed from a statistical perspective, a test is more uncertain if the DNN outputs similar probabilities for each class. The calibration issue also affects the uncertainty measurement (Guo et al. 2017). Some techniques such as Monte-Carlo Dropout (Gal and Ghahramani 2016) and Deep Ensemble (Lakshminarayanan, Pritzel, and Blundell 2017) have been proposed to calibrate the uncertainty. Different from the above works that proposed uncertainty metrics to measure model reliability, we utilize these metrics to estimate the capability of LLMs.

Benchmark

Problem Definition

Capability estimation for large language models is defined as predicting the ability (e.g., accuracy) of LLMs on unseen data without using label information (Hu et al. 2024).

Definition 1 (*Capability Estimation for LLM*)**.** Given a LLM M under test, a test set X_{test}, and a capability es-

Name	Type	Definition	Description
Average Training Performance (Average Train)	MMF	$P = \frac{1}{Num} \sum_{i=1}^{Num} P_{train}^{(i)}$	$P_{train}^{(i)}$: performance of i-th sample, Num : number of training data
Average Calibration Error (Average Confidence)	MMF confidence based	$P = \frac{1}{Num} \sum_{i=1}^{Num} Conf_{test}^{(i)}$	$Conf_{test}^{(i)}$: confidence of i-th sample, Num : number of test data
Confidence Threshold	MMF confidence based	$P = \frac{1}{Num} \sum_{i=1}^{Num} \mathbb{I}(Conf_{test}^{(i)} > threshold)$	$\mathbb{I}(condition)$ = 1 when $condition$ is True, else 0, Num : number of test data $threshold$: fixed confidence threshold
Temperature Scaling (TS)	MMF confidence based	$P = \frac{1}{Num} \sum_{i=1}^{Num} Conf_{test}^{(i),cal}$	$Conf_{test}^{(i),cal}$: calibrated confidence of i-th sample, Num : number of test data
Average Threshold Confidence (ATC (Garg et al. 2022))	MMF confidence based	$P = \frac{1}{Num} \sum_{i=1}^{Num} \mathbb{I}(Conf_{test}^{(i)} > threshold)$	$threshold$: threshold determined by meta-set
Confidence Threshold Count	MMB confidence based	feature: $\mathsf{Count}\left(Conf_x > threshold \mid x \in X_{meta}\right)$	Correlation between the high confident data number and model performance $threshold$: fixed confidence threshold
Token Frequency	MMB input based	feature: $\mathsf{ABS}\left(Token_{train} - Token_{meta}\right)$	Correlation between token frequency and model performance $Token$: token frequency histogram
Confidence Profile (Fu et al. 2023)	MMB confidence based	feature: $Conf$ of X_{meta}	Correlation between confidence scores and model performance
Deep Gini (Feng et al. 2020)	MMB uncertainty based	feature: uncertainty score (UC) of X_{meta} $UC_{deep} = \sum_{k \in K} p_k^2$	Correlation between uncertainty scores and model performance p_k : probability of k-th class, K : all classes
Margin	MMB uncertainty based	feature: $UC_{margin} = \max_{k \in K} p_k - \max_{k \in K \setminus \{k\}} p_k$	$k \in K \setminus \{k\}$: classes from K except k
Entropy	MMB uncertainty based	feature: $UC_{entropy} = -\sum_{k \in K} p_k \log p_k$	p_k : probability of k-th class, K : all classes
Model Differential Entropy MDE	MMB uncertainty based	feature: $UC_{MDE} = -\tau \cdot \log\left(\sum_{k \in K} e^{\frac{p_k}{\tau}}\right)$	τ : temperature (determined by the train data)

Table 1. Details of AutoEval methods.

timation method $E\left(M, X\right)$, capability estimation for LLM is the problem of $\mathsf{Min}\left|E\left(M, X_{test}\right) - \varrho\left(M, X_{test}, Y_{test}\right)\right|$ where Y_{test} is the label set corresponding to X_{test}, and $\varrho\left(M, X, Y\right)$ is a function measuring the performance of M when predicting the labels Y for the input set X.

Based on the construction, capability estimation methods can be divided into two groups, meta-model-free (MMF) methods and meta-model-based (MMB) methods. MMF indicates estimating the performance directly using extracted features, while MMB refers to learning the relationship between features and performance extracted from a set of data using meta-models.

Definition 2 (*Meta-Model-Free (MMF) Capability Estimation*)**.** Given a LLM M under test, a test set X_{test}, MMB extracts features F_{test} from M using X_{test}, and then estimate the performance of M directly using F_{test}.

Definition 3 (*Meta-Model-Based (MMB) Capability Estimation*)**.** Given a LLM M under test, a test set X_{test}, and meta-sets $X_{meta} = \left(X_{meta}^1, X_{meta}^2, ..., X_{meta}^N\right)$ and their corresponding labels Y_{meta} where N is the number of meta-sets, MMB extracts features F_{meta} and F_{test} from M using X_{meta} and X_{test}, and accesses the performance of M on X_{meta}, $P_{meta} = \varrho\left(M, X_{meta}, Y_{meta}\right)$. Then, MMB utilizes meta-model $M_{meta} : F \rightarrow P$ to learn the relationship between F_{meta} and P_{meta}, $\mathsf{Train}\left(M_{meta}, F_{meta}, P_{meta}\right)$. Finally, M_{meta} is used to estimate the capability of M on X_{test}, $P_{test} = M_{meta}\left(F_{test}\right)$.

Features (F) are important for both MMF and MMB methods and highly affect the precision of the estimation, meta-model M_{meta} is another important component for MMB methods. We consider three types of features, input-based features, confidence based features, and uncertainty-based features, and four types of meta-model Linear Regression, KNN, MLP, and XGBoost, in our benchmark.

Features

Prediction confidence is a basic feature for capability estimation. The calculation of prediction confidence in classification tasks and generation tasks is different, which can be summarized as follows:

The prediction confidence of classification tasks is calculated as the maximum softmax output probabilities:

$$Pro_{classification}^{k} = \frac{p_{y_k}}{\sum_{i \in K} p_{y_i}} \tag{1}$$

where K is the number of classes. For each choice class k, there should exist a corresponding probability value $Pro_{classification}^{k}$, and we regard the confidence as the highest probability (Kadavath et al. 2022), $Conf_{classification} = \mathsf{Max}\left(Pro_{classification}^{k} | k \in K\right)$.

The confidence of generation tasks is based on the negative log-likelihood $(\mathrm{NLL})^2$ of each output sequence as proposed by Fu *et al.* (Fu et al. 2023).

$$Conf_{generation} = -\sum_{t=1}^{|len(seq_k)|} \log p_{seq_k(t)} \tag{2}$$

where $p_{seq_k(t)}$ is output probability for t-th token in output sequence seq_k. Since the value $(\mathrm{NLL})^2$ does not have a unified scale, we normalize the score into the range $[0, 1]$.

In addition to confidence-based features, we propose to utilize the uncertainty score as another type of feature since it reflects the reliability of the model (Gal and Ghahramani 2016). Uncertainty features are based on the probability and confidence scores introduced above, and four types of uncertainty scores are covered in AEBench.

Table 1 presents feature details used for capability estimation methods. In total, AEBench supports 12 types of AutoEval methods where five are MMF methods and seven are MMB methods.

Meta Models

For each MMB method, we employ four representative meta models to learn the relationship between features and performance of LLMs, Linear Regression (Vapnik 1998), K-Nearest Neighbour (KNN) (Altman 1992), Multiple Layer Perceptron (MLP) (Hinton and Salakhutdinov 2006), and XGBoost (Chen and Guestrin 2016).

Linear Regression predicts outcomes based on a linear relationship between input variables and the target.

K-Nearest Neighbour (KNN) is a non-parametric method that estimates the performance based on the proximity to the nearest recorded data points. KNN is effective in scenarios where performance patterns are expected to be similar within localized clusters of data.

Multiple Layer Perceptron (MLP) is a type of neural network known for its ability to capture complex nonlinear relationships between features and performance. MLPs are useful for modeling intricate patterns in high-dimensional data.

XGBoost is an implementation of gradient boosted decision trees designed for speed and performance (Chen and Guestrin 2016). It is famous for its efficiency and effectiveness in a wide range of regression tasks, including performance estimation where feature interactions are complex.

Capability Measurements

Let y_{pred}, y_{truth}, and Num represent the model prediction, the ground truth, and the size of X_{test}, respectively.

Accuracy is a common measurement for classification tasks:

$$Accuracy = \frac{1}{Num} \sum_{i=1}^{Num} \mathbb{I}(y_{pred}^{(i)} = y_{truth}^{(i)}) \quad (3)$$

Recall is modified from the Recall metric used in classification tasks. It depends on the token set intersection between the predicted output and the ground truth output:

$$Recall = \frac{1}{Num} \sum_{i=1}^{Num} \frac{\sum_{j \in y_{truth}^{(i)}} \mathbb{I}(j \in y_{pred}^{(i)})}{|y_{truth}^{(i)}|} \quad (4)$$

Mean Absolute Error (MAE) is used to measure the difference between predicted performance ($P_{estimated}$), Accuracy or Recall described above, and the ground truth performance (P_{truth}).

$$MAE = |P_{estimated} - P_{truth}| \quad (5)$$

Experiments

Based on the benchmark, we conduct a comprehensive study to explore the following research questions:

- **RQ1:** *How does different singular feature affect the capability estimation?*
- **RQ2:** *How effective is feature combination in capability estimation?*
- **RQ3:** *How do capability estimation methods perform under out-of-distribution scenarios?*

Dataset	Original Size	Used Size	Task Type	Metric	Capability
CosmosQA	35600	10000	Classification	Accuracy	0.857
Halu Dialogue	10000	10000	Classification	Accuracy	0.653
Halu Summarization	10000	10000	Classification	Accuracy	0.607
HellaSwag	10000	10000	Classification	Accuracy	0.742
MMLU	10000	10000	Classification	Accuracy	0.613
IMDB	100000	10000	Classification	Accuracy	0.657
WILDS Amazon	10000	10000	Classification	Accuracy	0.621
TriviaQA Web	86447	10000	Extractive QA	Recall	0.706
TriviaQA Wikipedia	69881	10000	Extractive QA	Recall	0.656
Squad V2	92749	10000	Extractive QA	Recall	0.944

Table 2. Statistics of datasets.

Datasets and Base Model

We use *Llama3-8B* (instruct version, meta-llama/Meta-Llama-3-8B-Instruct) as the base model for all experiments. Additional results using DeepSeek models are available on the project website. We adopt the same hyper-parameters from Yuan *et al.* (Yuan et al. 2023), with a max generation length of 256 and the same prompt template under a zero-shot setting.

Ten datasets are considered in our experiments. The statistics of the datasets are shown in Table 2. CosmosQA (Huang et al. 2019) is a dataset for large-scale commonsense reasoning and reading comprehension, and it is formulated as multiple-choice questions. Halu Dialogue and Halu Summarization (Li et al. 2023) are about knowledge-grounded dialogue and text summarization from HaluEval. HellaSwag (Zellers et al. 2019) is a dataset for commonsense NLI, and MMLU (Massive Multitask Language Understanding) (Hendrycks et al. 2021) is designed for commonsense reasoning. IMDB (IMDB Movie Reviews) is a binary sentiment classification dataset. WILDS Amazon (Koh et al. 2021) is a multi-class sentiment classification dataset with 5 classes. We filter out the highest and lowest ratings and balance the ratio of them to 1:1 for binary classification. TriviaQA Web and TriviaQA Wikipedia are both extractive QA datasets from TriviaQA (Joshi et al. 2017), which is mainly about reading comprehension. Squad V2 (Rajpurkar, Jia, and Liang 2018) is the upgraded version of Squad, and it is also an extractive QA dataset.

We use $Accuracy$ as the performance metric for all classification tasks, and $Recall$ for extractive QA tasks because $Recall$ is better for the extractive QA tasks under open-ended generation. Among classification datasets, datasets with original size exceeding 10k are randomly sampled out of 10k in alignment with other datasets.

Experiment Settings

We use the Pearson correlation coefficient to analyze the relationship between features (F) and performance (P). We follow previous works (Fu et al. 2023; Yuan et al. 2023) to set hyperparameters. Euclidean distance is selected as the distance metric for KNN, and the number of neighbors is searched from 1 to 10. The MLP is a two-layer feedforward neural network with 1536 hidden units and a ReLU activation function, optimized with Adam optimizer and MSELoss. We use RandomizedSearchCV to search for the best hyper-parameters for XGBoost. The input of LLMs is with the zero-shot setting, and the output token and logits are kept for further analysis. Prompt examples can be found at

our project site. All experiments in this study are conducted on a computer with a 2.6 GHz Intel Xeon Gold 6132 CPU with a 32G NVIDIA Tesla V100 SXM2 GPU.

Results

RQ1. Singular Feature Analysis

First, we explore the effectiveness of using a single feature for capability estimation. The quantified mean absolute error (MAE) is used to evaluate the estimation performance, and the results are summarized in Table 3.

MMF methods comparison. Comparing each MMF method, the results demonstrate that Average Training (Avg Train) consistently outperforms other methods across all datasets. It is reasonable that since the distribution of training data is similar to test data, the mean performance on the training data offers a reliable estimate for unseen data. In contrast, the Average Confidence (Avg Confidence) feature exhibits the worst performance, with an average MAE of 0.2866. Such a high estimation error rate implies that LLMs are often overconfident, which confirms the overconfident findings in previous studies (Nguyen, Yosinski, and Clune 2015). Meanwhile, temperature scaling also shows bad performance which indicates the LLMs' overconfidence issue cannot be well calibrated through temperature scaling.

MMB methods comparison. Then, we compare each MMB method and observe that Linear Regression and KNN are better choices for meta-models where almost all the best results are achieved by these two models. Interestingly, only the Confidence Profile feature works well with MLP with a relatively small MAE compared to other features. This means simple MLP makes it hard to learn the relation between features and model performance and is not suggested for MMB methods. Considering all the results, Conf Count 0.8, Uncertainty-Deep, and Uncertainty-Entropy Linear Regression achieved the best average results, even compared with MMF methods, with an MAE of 0.340. These results highlight that our proposed uncertainty-based MMB strategy is promising for estimating the capability of LLMs.

Besides, we further analyze the impact of feature selection on MMB methods by investigating the correlation between features and LLM capability. Pearson correlation coefficient is used on Linear Regression-related results. The results demonstrated that uncertainty-related features have a strong correlation with performance. It is important to note that in certain domains like Summarization, IMDB, Wikipedia, and Squad, no single feature shows a notable correlation, suggesting these features fail to capture enough performance-related information in those specific datasets and there is still room for improvement.

Answer to RQ1: Our proposed uncertainty features-based MMB methods perform the best among all capability estimation methods. Linear Regression and KNN are more suitable meta-models for MMB methods.

RQ2. Feature Combination

As MMB methods can accept multiple features as input, we use our benchmark to analyze whether we can boost capability estimation methods using feature combinations. We use Linear Regression as a meta-model, as RQ1 revealed its superiority.

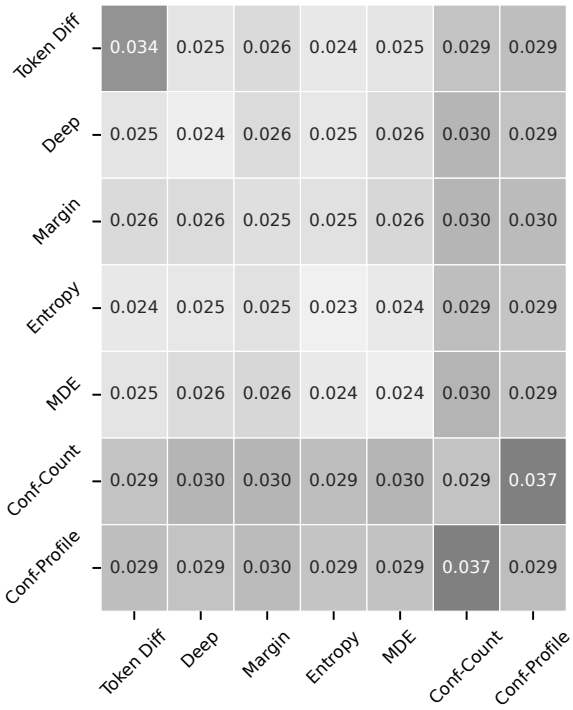

Figure 1: MAE of two-feature combinations on CosmosQA.

Binary Feature Combination. Figure 1 depicts the results of a combination of two features on the CosmosQA dataset. The results show that compared to single feature-based methods (results in Table 3), in most situations (31 out of 49 cases), feature combination cannot improve the estimation effectiveness. Only for Token Difference, by combining with others, the MAE is decreased with better performance, at most from 0.0344 to 0.0244 with Uncertainty Entropy. In summary, simply combining two features for MMB methods cannot enhance the capability estimation performance.

Multiple Feature Combination. Furthermore, we experiment with combinations among all features with a length ranging from two to seven. The results of all 120 combinations are shown in Table 4. We can see that the most effective combination is the mix of Uncertainty Margin, Uncertainty Entropy, and Uncertainty MDE, which collectively achieved the lowest average MAE of 0.0336. This combination not only excelled in classification contexts but also demonstrated robust efficacy across extractive QA datasets, validating the reliability and versatility of Uncertainty-based features again. Moreover, all leading combinations in Table 4 predominantly contain only uncertainty-based features, consistently outperforming they are the best singular features. In contrast, combinations involving the Confidence Profile generally result in dramatic MAE increases in both classification and extractive QA datasets, which are listed at the bottom of Table 4. Similar degradation trends are observed in combination with Token Difference. Through sys-

Feature	Datasets										Average
	Cosmos	Dialogue	Summarization	HellaSwag	MMLU	IMDB	Amazon	Web	Wikipedia	Squad	
MMF											
Avg Train	0.0339	0.0518	0.0373	0.0404	0.0315	0.0292	0.0369	0.0381	0.0367	0.0138	0.0350
Avg Confidence	0.0605	0.1741	0.3454	0.1319	0.1779	0.2277	0.2590	0.4281	0.3842	0.6768	0.2866
Conf Threshold 0.8	0.0415	0.0495	0.2935	0.0340	0.0555	0.1150	0.1230	0.5832	0.6036	0.6774	0.2576
Temperature Scaling	0.0242	0.0519	0.1462	0.0338	0.0339	0.0628	0.0882	0.3709	0.3429	0.7664	0.1921
ATC	0.0309	0.0528	0.0349	0.0320	0.0431	0.0475	0.0512	0.0578	0.0604	0.0278	0.0438
MMB, Meta Model = Linear Regression											
Confidence Profile	0.0294	0.0587	0.0396	0.0346	0.0418	0.0267	0.0512	0.0444	0.0478	0.0141	0.0388
Conf Count 0.8	0.0288*	0.0491*	0.0377	0.0363*	0.0344*	0.0291	0.0358	0.0381	0.0367	0.0142	0.0340
Token Difference	0.0344	0.0500	0.0377	0.0396	0.0318	0.0315	0.0427*	0.0381	0.0369	0.0145	0.0357
Uncertainty-MDE	0.0239*	0.0496*	0.0382	0.0348*	0.0360*	0.0286	0.0386*	0.0395*	0.0367	0.0147	0.0341
Uncertainty-Deep	0.0239*	0.0497*	0.0382	0.0349*	0.0361*	0.0286	0.0386*	0.0387	0.0367	0.0147	0.0340
Uncertainty-Margin	0.0255*	0.0487*	0.0380	0.0351*	0.0369*	0.0286	0.0388*	0.0396	0.0351	0.0142	0.0341
Uncertainty-Entropy	0.0234*	0.0505*	0.0385	0.0350*	0.0362*	0.0289	0.0384*	0.0379	0.0366	0.0145	0.0340
MMB, Meta Model = KNN											
Confidence Profile	0.0266	0.0513	0.0336	0.0372	0.0326	0.0268	0.0421	0.0399	0.0350	0.0154	0.0341
Conf Count 0.8	0.0339	0.0522	0.0366	0.0372	0.0305	0.0292	0.0502	0.0419	0.0390	0.0140	0.0365
Token Difference	0.0377	0.0741	0.0615	0.0430	0.0382	0.0310	0.0494	0.0475	0.0427	0.0165	0.0442
Uncertainty-MDE	0.0299	0.0500	0.0470	0.0387	0.0418	0.0360	0.0495	0.0453	0.0362	0.0193	0.0394
Uncertainty-Deep	0.0291	0.0536	0.0473	0.0368	0.0468	0.0345	0.0389	0.0406	0.0379	0.0169	0.0382
Uncertainty-Margin	0.0274	0.0454	0.0336	0.0347	0.0377	0.0395	0.0397	0.0372	0.0426	0.0163	0.0354
Uncertainty-Entropy	0.0251	0.0547	0.0433	0.0408	0.0393	0.0242	0.0482	0.0490	0.0362	0.0135	0.0374
MMB, Meta Model = MLP											
Confidence Profile	0.0319	0.0505	0.0419	0.0400	0.0319	0.0338	0.0400	0.2973	0.3464	0.0561	0.0970
Conf Count 0.8	0.8611	0.6894	0.5927	0.7499	0.6070	0.6749	0.6063	0.6893	0.6703	0.9423	0.7083
Token Difference	0.8587	0.6897	0.5911	0.7445	0.6047	0.6736	0.6104	0.6817	0.6705	0.9440	0.7069
Uncertainty-MDE	0.8613	0.6905	0.5921	0.7496	0.6070	0.6744	0.6051	0.6602	0.6167	1.0003	0.7057
Uncertainty-Deep	0.8613	0.6900	0.5922	0.7497	0.6077	0.6745	0.6046	0.9927	1.0174	1.0033	0.7793
Uncertainty-Margin	0.8613	0.6899	0.5924	0.7499	0.6065	0.6745	0.6039	0.6420	0.6098	1.0032	0.7033
Uncertainty-Entropy	0.8574	0.6882	0.5927	0.7499	0.6113	0.6745	0.6039	0.6839	0.6703	0.9426	0.7074
MMB, Meta Model = XGBoost											
Confidence Profile	0.0339	0.0519	0.0375	0.0387	0.0367	0.0305	0.0369	0.0381	0.0367	0.0138	0.0355
Conf Count 0.8	0.0337	0.0486	0.0372	0.0403	0.0315	0.0296	0.0384	0.0384	0.0367	0.0139	0.0348
Token Difference	0.0339	0.0507	0.0371	0.0352	0.0315	0.0295	0.0410	0.0396	0.0367	0.0141	0.0349
Uncertainty-MDE	0.0339	0.0510	0.0370	0.0406	0.0300	0.0293	0.0368	0.0377	0.0367	0.0141	0.0347
Uncertainty-Deep	0.0287	0.0506	0.0368	0.0398	0.0364	0.0299	0.0373	0.0386	0.0367	0.0140	0.0349
Uncertainty-Margin	0.0298	0.0470	0.0364	0.0405	0.0315	0.0297	0.0366	0.0385	0.0367	0.0145	0.0341
Uncertainty-Entropy	0.0337	0.0492	0.0371	0.0366	0.0349	0.0295	0.0404	0.0385	0.0367	0.0139	0.0351

Table 3. MAE of each method with a singular feature. Gray cells mark the best per dataset. * means the feature–performance correlation is significant (P-value < 0.05) for linear regression models.

tematic exploration, we can conclude that uncertainty-based features are more reliable and consistent in assessing model performance. Features like Confidence Profile and Token Difference are less effective in capturing information from data and conflict with other features.

Answer to RQ2: Margin, Entropy, and MDE achieve the best MAE (0.0336), highlighting the effectiveness of uncertainty metrics in estimating LLM capability.

RQ3. Estimation Under Out-of-Distribution

In contrast to the above experiments under the in-distribution hypothesis, most real-world applications present a scenario with train and test data from different distributions. To evaluate under more challenging OOD conditions, we designed experiments in which models were trained on one dataset and tested on another.

As expected, the estimation errors generally increase under OOD conditions than before. Notably, some exceptions illustrate the potential impact of domain similarity on model performance. For instance, when transferring from IMDB to Amazon within the same sentiment analysis domain, the MAE decreases, suggesting that similar content characteristics may mitigate the typical performance degradation seen under OOD conditions. There is also an MAE decrease from MMLU to Dialogue, from 0.0454 to 0.0335, where both datasets could share underlying task characteristics. In addition, the most pronounced estimation error occurs when transferring from Cosmos to Summarization and from Squad to Wikipedia on two types of tasks, with an increase in MAE of 740% and 746%, respectively. These findings showcase that the resemblance between in-distribution data and OOD data is a pivotal factor that affects the capability estimation. Among the evaluated meta-models, Linear Regression still stood out consistently for its superior performance in OOD scenarios. This was reflected in the count of best MAE values achieved, with 20 best values from Linear Regression compared to the second-best count of 8 from ATC.

Answer to RQ3: Classification tasks are less sensitive to distribution shifts than generation tasks, while uncertainty-based MMB methods exhibit greater robustness.

Discussion

Feature Analysis

Through our above study about the impact of features on capability estimation, LLMs frequently display a tendency

Feature Combination	Dataset Type		Average
	Classification	Extractive QA	
UC-Margin+UC-Entropy+UC-MDE	0.0354	0.0297	0.0336
UC-Deep+UC-Margin+UC-Entropy	0.0354	0.0304	0.0339
UC-Entropy+UC-MDE	0.0354	0.0304	0.0339
UC-Deep+UC-Entropy	0.0355	0.0303	0.0339
UC-Margin+UC-Entropy	0.0359	0.0295	0.0340
UC-Deep+UC-MDE	0.0356	0.0303	0.0340
UC-Deep+UC-Margin+UC-Entropy+UC-MDE	0.0360	0.0294	0.0340
UC-Deep+UC-Margin+UC-MDE	0.0360	0.0295	0.0341
UC-Margin+UC-MDE	0.0361	0.0299	0.0342
UC-Deep+UC-Entropy+UC-MDE	0.0359	0.0303	0.0342
...			
Token Difference+UC-MDE+Confidence-Threshold+Confidence-Profile	0.0429	0.0349	0.0405
Token Difference+UC-Deep+UC-Margin+UC-Entropy+UC-MDE+Confidence-Threshold+Confidence-Profile	0.0430	0.0349	0.0405
Token Difference+UC-Deep+UC-Margin+Confidence-Threshold+Confidence-Profile	0.0429	0.0350	0.0406
Token Difference+UC-Deep+UC-Margin+UC-Entropy+Confidence-Threshold+Confidence-Profile	0.0429	0.0352	0.0406
Token Difference+UC-Deep+UC-Margin+UC-MDE+Confidence-Threshold+Confidence-Profile	0.0431	0.0348	0.0406
Token Difference+UC-Margin+UC-Entropy+UC-MDE+Confidence-Threshold+Confidence-Profile	0.0430	0.0351	0.0406
Token Difference+UC-Deep+Confidence-Threshold+Confidence-Profile	0.0428	0.0355	0.0406
Token Difference+UC-Entropy+Confidence-Threshold+Confidence-Profile	0.0426	0.0363	0.0407
Token Difference+Confidence-Threshold+Confidence-Profile	0.0430	0.0354	0.0407
Confidence-Threshold+Confidence-Profile	0.0507	0.0355	0.0462

Table 4. Estimation performance of multiple feature combination on linear regression in ascending order by their average MAE.

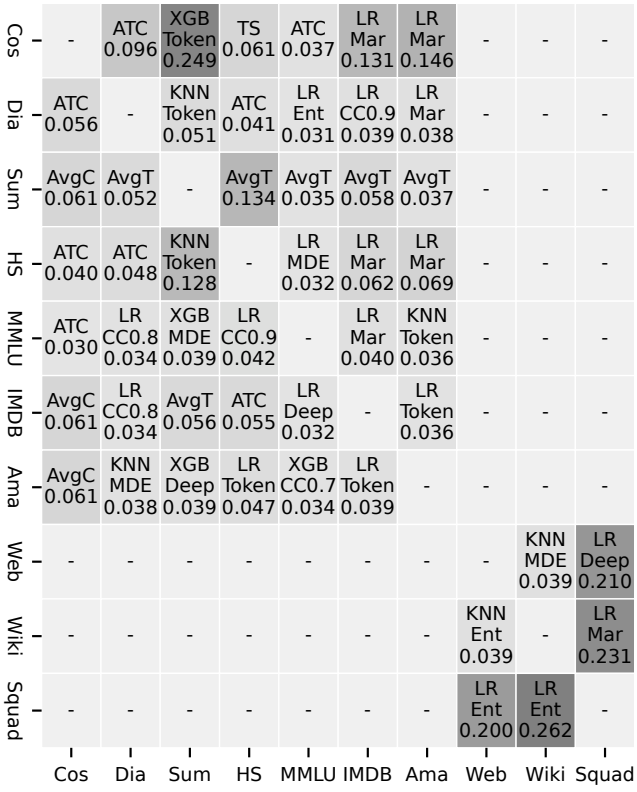

Figure 2: MAE on OOD scenarios. Y-axis: ID; X-axis: OOD. Datasets use first 3 letters (e.g., **Cos**=Cosmos). Methods are abbreviated (e.g., **XGB**=XGBoost, **AvgC**=AvgConf, **CC***=confidence count variants, **Ent**=Entropy).

towards overconfidence, leading to underperformance in confidence based features such as Average Confidence in most cases. Temperature Scaling and Average Threshold Confidence can help confidence calibration for estimation, but are much less effective than uncertainty-based features. Uncertainty-related features show a robust correlation with performance metrics and achieve lower estimation errors against other features across a variety of tasks. Combining multiple features can enhance the estimation, but not always, and combinations of uncertainty-based features generally perform the best.

Meta Model Analysis

After evaluating the performance of different meta-models, Linear Regression, K-Nearest Neighbors (KNN), and Extreme Gradient Boosting (XGBoost) demonstrate comparably lower MAE values, outperforming the best MMF results. Considering training cost, Linear Regression is the most efficient and effective model for capability estimation, and it consistently outperforms other models in most cases.

Conclusion and Future Work

In this work, we introduced the first benchmark for LLM capability estimation, AEBench, supporting more than ten capability estimation methods, including our uncertainty-based MMB methods. Using AEBench, we conducted a comprehensive study and revealed that uncertainty-based MMB methods perform the best with the lowest estimation error. We believe that our work provides new baselines and facilitates the direction of AutoEval in the LLM era. AEBench is simple and easy to extend. In the future, we plan to broaden the study with wider features and meta-models, thereby refining the capability estimation.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. U24A6009), the Luxembourg National Research Fund (Grant BRIDGES/2022/1S/17437536/TIMELESS), and Tianjin University–Lanzhou Jiaotong University Independent Research Fund (Grant No. 2025XSU-0018).

References

Altman, N. S. 1992. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3): 175–185.

Bao, S.; Sha, C.; Chen, B.; Peng, X.; and Zhao, W. 2023. In Defense of Simple Techniques for Neural Network Test Case Selection. In Just, R.; and Fraser, G., eds., *Proceedings of the 32nd ACM SIGSOFT International Symposium on Software Testing and Analysis, ISSTA 2023, Seattle, WA, USA, July 17-21, 2023*, 501–513. ACM.

Chen, M.; Tworek, J.; Jun, H.; Yuan, Q.; Pinto, H. P. D. O.; Kaplan, J.; Edwards, H.; Burda, Y.; Joseph, N.; Brockman, G.; et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Chen, T.; and Guestrin, C. 2016. XGBoost: A Scalable Tree Boosting System. In Krishnapuram, B.; Shah, M.; Smola, A. J.; Aggarwal, C. C.; Shen, D.; and Rastogi, R., eds., *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, 785–794. ACM.

Deng, W.; Gould, S.; and Zheng, L. 2021. What Does Rotation Prediction Tell Us about Classifier Accuracy under Varying Testing Environments? In Meila, M.; and Zhang, T., eds., *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, 2579–2589. PMLR.

Deng, W.; Suh, Y.; Gould, S.; and Zheng, L. 2023. Confidence and Dispersity Speak: Characterizing Prediction Matrix for Unsupervised Accuracy Estimation. In Krause, A.; Brunskill, E.; Cho, K.; Engelhardt, B.; Sabato, S.; and Scarlett, J., eds., *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, 7658–7674. PMLR.

Deng, W.; and Zheng, L. 2021. Are Labels Always Necessary for Classifier Accuracy Evaluation? In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, 15069–15078. Computer Vision Foundation / IEEE.

Der Kiureghian, A.; and Ditlevsen, O. 2009. Aleatory or epistemic? Does it matter? *Structural safety*, 31(2): 105–112.

Feng, Y.; Shi, Q.; Gao, X.; Wan, J.; Fang, C.; and Chen, Z. 2020. DeepGini: prioritizing massive tests to enhance the robustness of deep neural networks. In Khurshid, S.; and Pasareanu, C. S., eds., *ISSTA '20: 29th ACM SIGSOFT International Symposium on Software Testing and Analysis, Virtual Event, USA, July 18-22, 2020*, 177–188. ACM.

Fu, H. Y.; Ye, Q.; Xu, A.; Ren, X.; and Jia, R. 2023. Estimating Large Language Model Capabilities without Labeled Test Data. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, 9530–9546. Association for Computational Linguistics.

Gal, Y.; and Ghahramani, Z. 2016. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In Balcan, M.; and Weinberger, K. Q., eds., *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, 1050–1059. JMLR.org.

Garg, S.; Balakrishnan, S.; Lipton, Z. C.; Neyshabur, B.; and Sedghi, H. 2022. Leveraging unlabeled data to predict out-of-distribution performance. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Guo, C.; Pleiss, G.; Sun, Y.; and Weinberger, K. Q. 2017. On Calibration of Modern Neural Networks. In Precup, D.; and Teh, Y. W., eds., *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, 1321–1330. PMLR.

Hastie, T.; Friedman, J. H.; and Tibshirani, R. 2001. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer. ISBN 978-1-4899-0519-2.

Hendrycks, D.; Burns, C.; Basart, S.; Zou, A.; Mazeika, M.; Song, D.; and Steinhardt, J. 2021. Measuring Massive Multitask Language Understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Hendrycks, D.; and Gimpel, K. 2017. A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Hinton, G. E.; and Salakhutdinov, R. R. 2006. Reducing the dimensionality of data with neural networks. *science*, 313(5786): 504–507.

Hu, Q.; Guo, Y.; Xie, X.; Cordy, M.; Ma, L.; Papadakis, M.; and Le Traon, Y. 2024. Test optimization in DNN testing: a survey. *ACM Transactions on Software Engineering and Methodology*, 33(4): 1–42.

Hu, Q.; Guo, Y.; Xie, X.; Cordy, M.; Papadakis, M.; Ma, L.; and Traon, Y. L. 2023. Aries: Efficient Testing of Deep Neural Networks via Labeling-Free Accuracy Estimation. In *45th IEEE/ACM International Conference on Software Engineering, ICSE 2023, Melbourne, Australia, May 14-20, 2023*, 1776–1787. IEEE.

Huang, L.; Bras, R. L.; Bhagavatula, C.; and Choi, Y. 2019. Cosmos QA: Machine Reading Comprehension with Contextual Commonsense Reasoning. In Inui, K.; Jiang, J.; Ng, V.; and Wan, X., eds., *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, 2391–2401. Association for Computational Linguistics.

Joshi, M.; Choi, E.; Weld, D. S.; and Zettlemoyer, L. 2017. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In Barzilay, R.; and Kan, M., eds., *Proceedings of the 55th Annual Meeting of the*

Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers, 1601–1611. Association for Computational Linguistics.

Kadavath, S.; Conerly, T.; Askell, A.; Henighan, T.; Drain, D.; Perez, E.; Schiefer, N.; Hatfield-Dodds, Z.; DasSarma, N.; Tran-Johnson, E.; Johnston, S.; Showk, S. E.; Jones, A.; Elhage, N.; Hume, T.; Chen, A.; Bai, Y.; Bowman, S.; Fort, S.; Ganguli, D.; Hernandez, D.; Jacobson, J.; Kernion, J.; Kravec, S.; Lovitt, L.; Ndousse, K.; Olsson, C.; Ringer, S.; Amodei, D.; Brown, T.; Clark, J.; Joseph, N.; Mann, B.; McCandlish, S.; Olah, C.; and Kaplan, J. 2022. Language Models (Mostly) Know What They Know. *CoRR*, abs/2207.05221.

Koh, P. W.; Sagawa, S.; Marklund, H.; Xie, S. M.; Zhang, M.; Balsubramani, A.; Hu, W.; Yasunaga, M.; Phillips, R. L.; Gao, I.; Lee, T.; David, E.; Stavness, I.; Guo, W.; Earnshaw, B.; Haque, I. S.; Beery, S. M.; Leskovec, J.; Kundaje, A.; Pierson, E.; Levine, S.; Finn, C.; and Liang, P. 2021. WILDS: A Benchmark of in-the-Wild Distribution Shifts. In Meila, M.; and Zhang, T., eds., *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, 5637–5664. PMLR.

Lakshminarayanan, B.; Pritzel, A.; and Blundell, C. 2017. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. In Guyon, I.; von Luxburg, U.; Bengio, S.; Wallach, H. M.; Fergus, R.; Vishwanathan, S. V. N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 6402–6413.

Li, J.; Cheng, X.; Zhao, X.; Nie, J.; and Wen, J. 2023. HaluEval: A Large-Scale Hallucination Evaluation Benchmark for Large Language Models. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, 6449–6464. Association for Computational Linguistics.

Liu, J.; Xia, C. S.; Wang, Y.; and Zhang, L. 2024. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. *Advances in Neural Information Processing Systems*, 36.

Maggio, S.; Bouvier, V.; and Dreyfus-Schmidt, L. 2022. Performance Prediction Under Dataset Shift. In *26th International Conference on Pattern Recognition, ICPR 2022, Montreal, QC, Canada, August 21-25, 2022*, 2466–2474. IEEE.

Nguyen, A. M.; Yosinski, J.; and Clune, J. 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, 427–436. IEEE Computer Society.

Niculescu-Mizil, A.; and Caruana, R. 2005. Predicting good probabilities with supervised learning. In Raedt, L. D.; and Wrobel, S., eds., *Machine Learning, Proceedings of the Twenty-Second International Conference (ICML 2005), Bonn, Germany, August 7-11, 2005*, volume 119 of *ACM International Conference Proceeding Series*, 625–632. ACM.

Rajpurkar, P.; Jia, R.; and Liang, P. 2018. Know What You Don't Know: Unanswerable Questions for SQuAD. In Gurevych, I.; and Miyao, Y., eds., *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, 784–789. Association for Computational Linguistics.

Saxena, R.; Kim, T.; Mehra, A.; Baek, C.; Kolter, J. Z.; and Raghunathan, A. 2024. Predicting the Performance of Foundation Models via Agreement-on-the-Line. In Globersons, A.; Mackey, L.; Belgrave, D.; Fan, A.; Paquet, U.; Tomczak, J. M.; and Zhang, C., eds., *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Scheffer, T.; Decomain, C.; and Wrobel, S. 2001. Active Hidden Markov Models for Information Extraction. In Hoffmann, F.; Hand, D. J.; Adams, N. M.; Fisher, D. H.; and Guimarães, G., eds., *Advances in Intelligent Data Analysis, 4th International Conference, IDA 2001, Cascais, Portugal, September 13-15, 2001, Proceedings*, volume 2189 of *Lecture Notes in Computer Science*, 309–318. Springer.

Snoek, J.; Ovadia, Y.; Fertig, E.; Lakshminarayanan, B.; Nowozin, S.; Sculley, D.; Dillon, J. V.; Ren, J.; and Nado, Z. 2019. Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift. In Wallach, H. M.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E. B.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 13969–13980.

Vapnik, V. 1998. *Statistical learning theory*. Wiley. ISBN 978-0-471-03003-4.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In Guyon, I.; von Luxburg, U.; Bengio, S.; Wallach, H. M.; Fergus, R.; Vishwanathan, S. V. N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 5998–6008.

Yuan, L.; Chen, Y.; Cui, G.; Gao, H.; Zou, F.; Cheng, X.; Ji, H.; Liu, Z.; and Sun, M. 2023. Revisiting Out-of-distribution Robustness in NLP: Benchmarks, Analysis, and LLMs Evaluations. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Zellers, R.; Holtzman, A.; Bisk, Y.; Farhadi, A.; and Choi, Y. 2019. HellaSwag: Can a Machine Really Finish Your Sentence? In Korhonen, A.; Traum, D. R.; and Màrquez, L., eds., *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, 4791–4800. Association for Computational Linguistics.