Analyzing Open LLM Performance

Team MMA - Mohamed Bakr, Marc Kleyman, Ankit Dey

2023-10-19

Contents

Introduction

With the release of ChatGPT and the surge in AI adoption, many companies are searching for feasible large language model solutions to solve their problems. Many small-cap and mid-cap companies are unable to dedicate the resources necessary to work on this endeavor themselves, so they are seeking help from some of the top consulting firms in the nation. An industry research study performed by Future Market Insights shows that the AI Consulting industry has a market size of roughly $525 billion in 2023[1] and will most likely continue to grow.

As employees of one of the leading AI Consulting firms, our team of machine learning consultants was tasked with building a large language model for a mid-cap customer that we want to do recurring business with. To ensure our firm was showcasing its best work, we decided to conduct this study to determine which model was the best and if the tuning type was the key to its success (for replicability). To explore this idea, we formulated the following research question:

> *"How does changing the tuning type of a Large Language Model (LLM) cause a change in the quality of its outputs?"*

Below is our initial causal path diagram for our research:

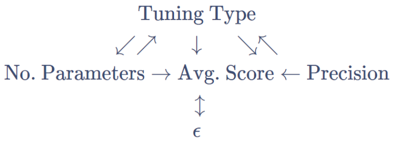

Figure 1: Initial Causal Path Diagram

Data and Methodology

In our analysis, we utilize Open LLM Performance Benchmark data[2], which is a survey of 1472 Comprehensive Language Model Evaluation Metrics that is updated from the HuggingFace leader board[3] as of October 16th, 2023. HuggingFace is a platform and open-source provider of machine learning technologies. It allows users to share datasets and machine learning models, and showcase their work. The HuggingFace Open LLM Leaderboard tracks and ranks evaluations of LLMs using benchmarks from the Eleuther AI Language Model Evaluation Harness. The unit of observation in this dataset is one singular LLM. The goal of this analysis is to determine which LLM is best for inference and are varying tuning `Type` (X) to have a positive impact on `Average Score` (Y). We can only present one model to the client, so we cannot use multiple models or parts of a model as our unit of observation.

We performed all exploration and model building on a 70% subsample of the data. The remaining 30%, totaling 441 rows, was used to generate the statistics in this report.

Tuning type is operationalized by the X variable `Tuning Type`, which is a nominal variable with four classes: Instruction-Tuned, Fine-Tuned, Pretrained, and Reinforcement Learning-Tuned. The X concept refers to the weight calibration method that was used to guide the large language model's training after it was pretrained on a dataset. `Tuning Type` was one-hot encoded to calculate correlations and allow for identification of OVB. Model quality is operationalized by the Y variable `Average score`, which is an average score (between 1 and 100) across four benchmarks that measure different aspects of the quality of an LLM's outputs. Each LLM goes through all four benchmarks and the benchmarks each measure a different aspect of the quality of the LLM's outputs: `ARC` (question-answering abilities), `HellaSwag` (common sense reasoning), `MMLU` (language understanding), and `TruthfulQA` (truthfulness). The scores in each benchmark are metric and continuous and are determined by giving the large language model a series of questions and calculating the percentage

[1] Global AI Consulting Services Market Outlook (2023 to 2033).Future Market Insights

[2] Open LLM Performance Benchmark.Open LLM performance Dataset

[3] Hugging Face Leaderboard. Open LLM Leaderboard

of questions the model gets correct. The average score for each LLM is the simply the mean of the four percentage scores.

Table 1: Accounting Table

Cause	Number of Samples Available For Analysis (after removal for cause)	Number of Samples Removed
Start	1472	NA
Missing Tuning Type	1469	3
Exploration subsample	441	1028

As we report in Table 1, we removed 3 models with missing `Type`, leaving 1469 models. Each row represents a single LLM. Each LLM is evaluated on the test dataset of each benchmark, and the residuals are aggregated within each benchmark to compute the score for that benchmark for that LLM. There are 1420 unique LLMs in the dataset, with a 43 LLMs being benchmarked repeatedly.

We are interested in the effect of changing tuning type `Tuning Type` on the average score the model receives across all four benchmarks `Average Score`. The current scientific consensus suggests that the relationship between score and the number of parameters (the only metric co-variate that we consider) may be linear or close to linear within the range of parameters considered in the data. We therefore create regression models with no transformations applied to the output or predictor variables. In other words, we fit regressions of the form,

$$\widehat{Avg.Score} = \beta_0 + \beta_1 \cdot (Fine\ Tuned) + \beta_2 \cdot (Instruction\ Tuned) + \beta_3 \cdot (RL\ Tuned) + \mathbf{Z}\gamma$$

Where β_1 represents the predicted change in average score when changing from Pretrained to Fine Tuned, β_2 represents the predicted change in average score when changing from Pretrained to Instruction Tuned, β_3 represents the predicted change in average score when changing from Pretrained to RL-Tuned, $\mathbf{Z}$ is a row vector of additional covariates, and γ is a column vector of predicted coefficients for these covariates.

We considered specifications that included the score of each LLM on individual benchmarks (`ARC`, `HellaSwag`, `TruthfulQA`, or `MMLU`), however these are outcome variables that are directly related to average score `Average Score`, and are therefore not appropriate to include as predictors. We also considered specifications that include a nominal variable that represents the base model used prior to tuning, derived from the model name variable `model_name_for_query`. The method for deriving this variable required matching up pieces of the names of LLMs to identify LLMs with the same base model, which was unreliable and therefore not used. We explored two other covariates called `Params` (parameters) and `Precision`. `Params` is the number of estimated coefficients in the model, which is mathematically known to impact the score – LLMs with a higher number of `Params` usually perform better on benchmarks. The number of parameters may also be correlated with the tuning type because it may be too expensive to use certain tuning types on a model with more parameters.

For `Precision`, encoding floats as torch.bfloat16 (brain floats) over torch.float16 is known to be associated with an increase in inference accuracy, and therefore an increase in score, due to the increased range of brain floats (Grobbelaar, 2023). Additionally, precision type may be correlated with the tuning type – it is possible that some forms of tuning do not work well with the decreased range of regular 16-bit floats.

Results

All of the one hot encoded tuning categories were found to have statistically significant effects across all 3 models, supporting the idea that tuning type has a causal impact on score. The addition of number of parameters as a predictor in model 2 led to a statistically significant change in the regression coefficient for the fine-tuned category (T = -2.12, p = 0.041), representing a reduction in bias for that coefficient. The test F-statistic for model 2 (F = 54.61, p = 0) is greater than that of model 3 (F = 28.09, p = 0), indicating

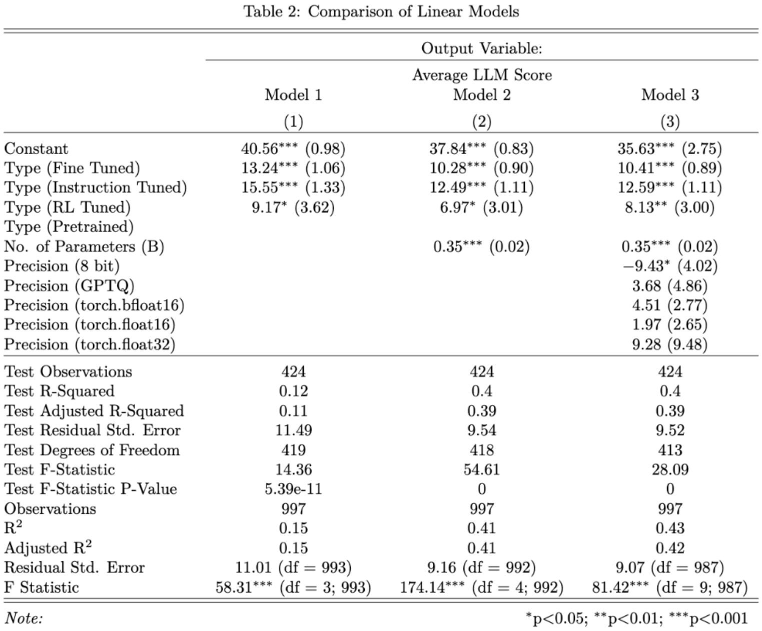

Table 2: Comparison of Linear Models

	Output Variable:		
		Average LLM Score	
	Model 1	Model 2	Model 3
	(1)	(2)	(3)
Constant	40.56*** (0.98)	37.84*** (0.83)	35.63*** (2.75)
Type (Fine Tuned)	13.24*** (1.06)	10.28*** (0.90)	10.41*** (0.89)
Type (Instruction Tuned)	15.55*** (1.33)	12.49*** (1.11)	12.59*** (1.11)
Type (RL Tuned)	9.17* (3.62)	6.97* (3.01)	8.13** (3.00)
Type (Pretrained)			
No. of Parameters (B)		0.35*** (0.02)	0.35*** (0.02)
Precision (8 bit)			−9.43* (4.02)
Precision (GPTQ)			3.68 (4.86)
Precision (torch.bfloat16)			4.51 (2.77)
Precision (torch.float16)			1.97 (2.65)
Precision (torch.float32)			9.28 (9.48)
Test Observations	424	424	424
Test R-Squared	0.12	0.4	0.4
Test Adjusted R-Squared	0.11	0.39	0.39
Test Residual Std. Error	11.49	9.54	9.52
Test Degrees of Freedom	419	418	413
Test F-Statistic	14.36	54.61	28.09
Test F-Statistic P-Value	5.39e-11	0	0
Observations	997	997	997
R^2	0.15	0.41	0.43
Adjusted R^2	0.15	0.41	0.42
Residual Std. Error	11.01 (df = 993)	9.16 (df = 992)	9.07 (df = 987)
F Statistic	58.31*** (df = 3; 993)	174.14*** (df = 4; 992)	81.42*** (df = 9; 987)
Note:			*p<0.05; **p<0.01; ***p<0.001

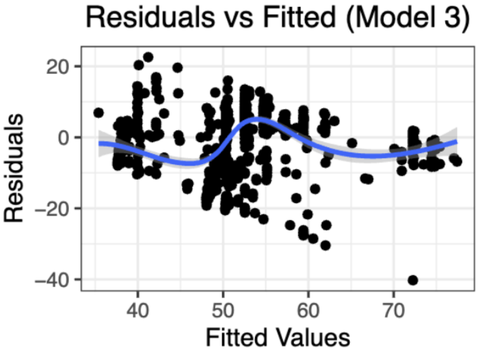

Figure 2: Regression Table and Residuals vs. Fitted Plot

that the addition of precision predictor variables in model 3 does not lead to a proportionately better fit. However, since at least one of the regression coefficients for precision was found to be statistically significant (8-bit precision, β_{8bit} = -9.43, p = 0.019), we recommend model 3 as a basis for estimating causal effects and making modeling decisions. 8-bit precision is the only precision category with a statistically significant coefficient. The negative effect of changing to 8-bit precision from another precision type on predicted average score is large compared to the positive effect of adding another billion parameters to the model (β_{8bit} = -9.43, β_{params} = 0.35), which makes sense since 8-bit precision covers the smallest range of numbers out of all of the precision types. For context, we would need to remove around 32.5 billion parameters from the model to produce the same negative effect on predicted average score as switching from torch.float16 to 8-bit precision, holding all else constant. Thus, we recommend against using 8-bit precision in the LLM to be developed for the client. The low p-value for β_{params} (p = 0) indicates a high degree of confidence in the positive impact of increasing the number of parameters on average score. To put this in context, we would need to add around 36 billion parameters to the model to produce the same positive impact on predicted average score as switching from a pre-training only strategy to instruction tuning, holding all else constant. Thus, we recommend developing a larger model for the client – 70 billion parameters is frequently chosen to maximize inference performance without being too costly, however, this should be adjusted based on use case, as some specific tasks may perform well with less parameters. The coefficients for the fine-tuned ($\beta_{fine-tuned}$ = 10.41, p = 0), instruction-tuned ($\beta_{instruction-tuned}$ = 12.59, p = 0), and RL-tuned ($\beta_{RL-tuned}$ = 8.13, p = 0.007) categories are all relatively large and positive, indicating a significant positive impact of tuning on average score as compared to only pre-training. The instruction-tuned category seems to have the greatest positive impact on average score since it has the largest coefficient with small standard error (Std. Error = 1.11), so we recommend the development team to use instruction tuning or a combination of tuning strategies that includes instruction-based tuning to produce the best performing LLM for the client.

Limitations

Regarding IID assumptions, the identical distribution assumption may be challenged since the LLMs aren't required to share a common architecture. The independence assumption may be challenged since LLMs built by the same developer may have similar scores, but we removed any duplicate LLMs with the same model name. Overall, we can assume IID is violated.

The VIF for all coefficients were between 1 and 2 except for 16-bit floats and 16-bit bfloats, which have a VIF of roughly 9. This indicates that there is no multicollinearity except with 16-bit floats and 16-bit bfloats. Because the VIF is greater than 4, we can assume there is moderate collinearity.

To check for linear conditional expectation, we created a plot of residuals vs. predicted values (Figure 2). Because the line is nonlinear and away from zero, we can assume that the linear conditional expectation assumption is violated.

To check for normality, we plotted the residuals on a Q-Q plot and a histogram. Both show that the data is close to following a normal distribution, but slightly heavy-tailed. Therefore, we can assume that the normality assumption is satisfied.

A studentized Breusch-Pagan test was run to test for homoscedasticity. The result was a p-value of 0.04, meaning the model has evidence of heteroscedasticity and violates the homoscedasticity assumption.

To check for zero conditional mean, we created a plot of residuals vs fitted values. The data created a non-linear lowess smoothing line fluctuating from -8 to 5. Because this line is not horizontal and near 0, we can assume that the zero conditional mean assumption is violated.

The four LLM performance benchmark tests may not perfectly encapsulate what the "best model" is due to the types of questions asked, but we do not have access to these models' performance on other benchmark tests. We would like to have seen some benchmark tests reflecting quantitative reasoning through math or coding questions for a more comprehensive examination to determine the "best model". The pre-training data set for each model was not given and this omitted variable may have an impact on the score. We also considered using the model name as a potential covariate because it includes the underlying base model and may hint towards the architecture and pre-training data, but it may be difficult to parse that information with enough accuracy from the model name.

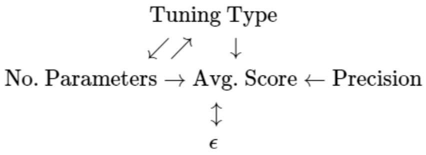

Figure 3: Updated Causal Path Diagram

Conclusion

Our research demonstrates the positive impact of all tuning types for optimizing model performance while Instruction Tuning shows the most significant impact. We recommend building a larger model with more parameters since increasing the number of parameters had a positive impact on the average score. Additionally, we recommend avoiding 8-bit precision due to its negative influence on the Average Score.

Moving forward, we believe exploring tuning strategies inspired by instruction or combined tuning strategies are promising avenues for future research.