Towards Interpretable Machine Reading Comprehension with Mixed Effects Regression and Exploratory Prompt Analysis

Anonymous ACL submission

Abstract

We investigate the properties of natural language prompts that determine their difficulty in machine reading comprehension (MRC) tasks. While much work has been done benchmarking language model (LM) performance at the task level, there is considerably less literature focused on how individual task items can enhance interpretability for MRC. We perform a mixed effects analysis on the behavior of three major LMs, comparing their performance on a large multiple choice MRC task to explain the relationship between predicted accuracy and different prompt features. First, we observe a divergence in LM accuracy as the prompt's token count grows with overall stronger LMs increasing in accuracy and overall weaker LMs decreasing. Second, all LMs exhibit consistent accuracy gains with increasing syntactic complexity. Third, a post hoc analysis revealed that the most difficult prompts had the greatest ability to discriminate between different LMs, suggesting their outsized usefulness in MRC evaluation methods.

1 Introduction

As of late, the research community has been fiercely debating whether recent developments in deep neural language modeling indicate true machine understanding of natural language or whether they merely demonstrate shallow mimicry of the patterns of language. Proponents of the understanding hypothesis have argued that the speed with which new, more difficult benchmarks must be created to keep up with advancement in pretrained LM capabilities suggests that human-level language comprehension is not far off (Wang et al., 2019). Devlin et al. (2019) state that "recent empirical improvements ... have demonstrated that rich, unsupervised pre-training is an integral part of many language understanding systems." Detractors have conversely argued that these benchmarks are not adequate indicators of real understanding because of their contrived and narrowly defined format, which does not generalize to human language as a whole (Niven and Kao, 2019; Zellers et al., 2020; Bender et al., 2021). While we cannot settle the MRC debate in the scope of this paper, we intend to lay the groundwork for deeper investigation into the methodology of MRC evaluation by answering the following question: "Which linguistic features predict LM accuracy on MRC tasks?" Answering this question will not only shed light on why LMs behave the way they do but also provide an opportunity to test an evaluation method that has not yet seen wide adoption in NLP research. To this end, we have conducted a narrow and deep investigation into LM performance on the RACE-h multiple choice MRC dataset (Lai et al., 2017), leveraging the advantages of mixed effects regression to enhance the interpretability our results, which indicate that this methodology can yield meaningful insights into the comparative behavior of different LMs beyond the capabilities of the simple benchmarking paradigm.

1.1 Why Mixed Effects?

Mixed effects models are statistical models that do not assume independent homoscedastic residuals (West et al., 2022). In other words, they are formulated to tolerate variables that are not necessarily randomly sampled and whose residuals may not have constant variance as is often the case with natural language corpora and benchmarking datasets (Baayen et al., 2008). Conversely, traditional fixed effects models require all samples to be independent and from an identical distribution (i.i.d. samples), making them generally inappropriate for such datasets. Datasets which lend themselves to a mixed effects analysis typically include those with clustered data or random block experimental designs as well as longitudinal or repeated measures sampling. Such data are inherently dependent between clusters, blocks, or repeated individual subjects. While mixed effects models have increasingly grown in popularity in the medical, biological, and social sciences for their flexibility and expressive power in hypothesis testing on complex datasets (West et al., 2022), they are still not widely used in the evaluation of LM performance or empirical NLP research more broadly (Riezler and Hagmann, 2022). Standard practice in LM evaluation still primarily relies on the train/dev/test paradigm, which satisfies i.i.d artificially by shuffling, partitioning, and cross-validating the dataset while often ignoring the statistically dependent structure of the data (Berg-Kirkpatrick et al., 2012). While this problem can be mitigated by averaging samples within blocks or clusters and fitting the model to the block averages, this workaround suffers from significant information loss and underestimates the amount of variation in the dataset (Baayen et al., 2008). Conversely, the mixed effects approach takes every data point and every grouping structure into account and can fully describe the variance both within and between groups via random intercepts and random slopes respectively. In addition, it can protect against inflated Type I error rates when fitting models to larger datasets, as was demonstrated by Baayen et al. (2008).

2 Related Work

Despite the tremendous growth in our ability to train and benchmark the performance of increasingly large LMs over the last ten years, our ability to analyze, contextualize, and understand their performance has not kept pace. This is because the most commonly used evaluation metrics have limited ability to help us understand *why* LMs behave the way they do, as "[authors] usually report a single high score of a model that has been trained with ... maximal hardware resources and maximal computational resources for extensive meta-parameter search" (Riezler and Hagmann, 2022). As such, Riezler and Hagmann (2022) recommend the use of linear mixed effects models for evaluating performance on NLP tasks, as they "allow us to estimate the variance induced by particular meta-parameter settings ... in a general way." Indeed, there seems to be a growing sense that current datasets and measurement techniques have become inadequate for the general task of LM evaluation, as Bowman and Dahl (2021) conclude that "benchmarking for NLU is broken" due to the lack of statistical validity and power of its techniques and the preponderance of inaccurate annotations in its datasets.

While some researchers have argued that evaluation methodologies should focus more on measuring the quality of natural language generation (NLG) rather than simple classification tasks (Zellers et al., 2020), NLG quality is notoriously difficult to define much less measure (Sai et al., 2020). As such, it has often been more practical for researchers to use simple metrics that frame NLG evaluation as a classification task, as Hendrycks et al. (2020) and Zellers et al. (2018) do. In addition, Zellers et al. (2018) and Zellers et al. (2019) use a technique known as adversarial filtering to improve the robustness of MCQA datasets to LMs that select answers based on shallow stylistic language patterns rather than grounded commonsense inference.

3 Methods

3.1 Dataset Collection

We used RACE-h – the popular MCQA-MRC dataset – which consists of 69,394 multiple choice questions (MCQs) collected from Chinese high school ESL examinations and comes prepartitioned into a 90/5/5 train/dev/test split, of which we chose to only use the test partition, leaving us with 3,498 MCQs arranged into 1,045 statistically independent clusters (Lai et al., 2017). Each MCQ has exactly four possible answer choices, and each cluster contains a single context passage shared between each MCQ in the cluster, making the individual MCQs statistically dependent, thus motivating the mixed effects analysis.

In the original publication of the dataset, Lai et al. (2017) use a sample of 250 MCQs to estimate the proportion of ill-formed MCQs at around 7.1%, though later analyses have suggested that it could be much higher. Zyrianova et al. (2023) take a more stringent and exhaustive approach to detecting errors in the dataset and reported that 61.5% of MCQs are unacceptable. This discrepancy is likely explainable by the highly divergent acceptability criteria used by the authors, though the true error rate in RACE-h is uncertain.

3.2 Language Model Inference

We posed each MCQ in the dataset to each of three major LMs in the GPT series – Davinci-002, Davinci-003, and GPT-4 via the OpenAI API – which are generally agreed to vary in overall task-level performance (Brown et al., 2020; OpenAI,

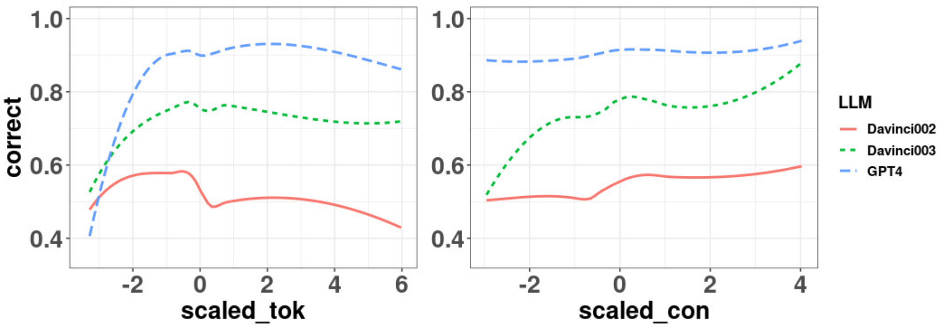

Figure 1: LOESS plots for token count and syntactic complexity predictors of probability correct. Continuous covariates are expressed in standard deviations from the mean.

2023). At inference time, we prompted each LM with temperature set to 0 and greedy search enabled for simplicity. After stripping peripheral white space from the response, we compared it to the gold response, which could only be “A”, “B”, “C”, or “D”. We then used these comparisons to create a binary response vector to fit each mixed effects model and frame the MCQA task as binary classification.

3.3 Linguistic Feature Measurement

For each MCQ, we measured two linguistic features that we expected could predict LM accuracy, thus providing an explanation for the behavior observed in the data collected. First, we measured the token count of each MCQ, reasoning that the varying context windows could affect an LM’s ability to comprehend longer prompts. Second, we devised a proxy for syntactic complexity by counting the number of nodes in the constituency parse tree generated by the Stanza parser (Qi et al., 2020) and dividing that by the token count to prevent multicollinearity. Finally, we scaled and centered each predictor so as to avoid convergence issues in model fitting. Additional information on our measurement procedures can be found in Appendix A.

Algorithm 1 Calculate Syntactic Complexity

function SCORE($sentence$, $parseTree$)
    $TC \leftarrow$ TOKENCOUNT($sentence$)
    $NC \leftarrow$ NODECOUNT($parseTree$)
    return NC/TC
end function

function NODECOUNT($node$)
    $count \leftarrow 1$
    for each $child$ in $node.children$ **do**
        $count \leftarrow count + $ NODECOUNT($child$)
    end for
    return $count$
end function

4 Regression Analysis

Because of the statistical dependence of MCQs within clusters, traditional regression models that assume statistical independence would be inappropriate for modeling performance on RACE-h. Instead, we used generalized linear mixed effects regression (GLMER), which allowed us to construct logistic regression models which account for each MCQ cluster as a separate random effect. This makes it possible to identify salient MCQ clusters for a more detailed post hoc analysis. It also allows us to estimate the fixed effects coefficients without loss of accuracy or the need to discard statistically dependent MCQs.

We used R (R Core Team, 2021) to preprocesses and conjoin all of the MCQs, responses, and linguistic measures and fit multiple GLMER models with the `lme4` package (Bates et al., 2015). We also used the `flexplot` package (Fife, 2023) for fitting LOESS models that were useful for visualizing and forming our preliminary intuitions about the data. The LOESS plots in Figure 1 suggest that very short prompts are harder for LMs to correctly answer in general, though this may just reflect data sparseness at the lower extreme. As prompts get longer, LM performance shoots up then either plateaus or falls off in the case of Davinci-002. A more interesting relationship can be seen in the second plot where accuracy grows almost monotonically for each of the three LMs. This observation runs contrary to the commonly accepted hypothesis in the psycholinguistic readability literature that greater syntactic complexity makes passages and questions harder to correctly answer (Eslami, 2014). Despite their great expressiveness, the LOESS plots do not provide parametric formulas or estimations of statistical significance for the observed relationships, meaning

$$\text{logit}\big(P_{ijk}\big) = \beta_0 + \beta_1 TOK_i + \beta_{2j} LLM_j + \beta_{3j}(TOK_i * LLM_j) + PSG_k \quad (1)$$

$$\text{logit}\big(P_{ijk}\big) = \beta_0 + \beta_1 CON_i + \beta_{2j} LLM_j + \beta_{3j}(CON_i * LLM_j) + PSG_k \quad (2)$$

$$P_{ijk} = P(COR_{ijk} = 1 \mid PSG_k) = {}^1\!/_{1+e^{-\text{logit}\left(P_{ijk}\right)}} \quad (3)$$

Figure 2: Specification of the mixed effects structure of the random intercepts GLMER models. The predicted log odds of a correct answer COR_{ijk} are given by Equations 1 and 2 where TOK_i is the scaled token count of the i_{th} MCQ; CON_i is the scaled syntactic complexity of MCQ_i; $LLM_j = 1$ for the j_{th} LLM or 0 for the mean LLM; and PSG_k is the k_{th} passage (i.e. k_{th} MCQ cluster). P_{ijk} can then be obtained using the inverse logit function for $i \in \{1, ..., 3498\}$, $j \in \{1, 2, 3\}$, and $k \in \{1, ..., 1045\}$.

they are not by themselves sufficient for drawing valid conclusions via null hypothesis significance testing.

We then fitted two GLMER models using the `glmer` function from `lme4`. The first specifies the token count and LM variables in addition to their interaction terms as the fixed effects predictors and random effects by MCQ passage (indicating a unique cluster). The second is much the same, except with the constituency complexity substituted for the token count. As can be seen in Appendix B, no convergence failure or singular fit was detected for either GLMER model. In addition, the `vif` function showed no multicolinearity between fixed effects parameters, indicating the validity and interpretability of the estimated fixed effects. Finally, a series of ablative likelihood ratio tests with the `anova` function showed that all of the predictors in both models contributed significantly to goodness of fit ($p < 0.05$), except for the `scaled_con:LLM` interaction terms.[1]

4.1 Fixed Effects Interpretation

Looking at the first regression table, we can see that the fixed effects coefficients for the population intercept, the offset for Davinci-002, and the interaction between token count and Davinci-002 were all statistically significant. The log odds estimate of the population intercept is 1.31 ($SE = 0.04$), corresponding to a probability of 0.79 for an MCQ with mean token count to be answered correctly assuming mean LM performance. The log odds offset for Davinci-002 is -1.12 ($SE = 0.04$), which corresponds to a probability of 0.55 ($= \sigma(1.31 - 1.12)$) for mean token count. The log odds offset for Davinci-003 is -0.04 ($SE = 0.04$), which makes the corresponding probability 0.78 ($= \sigma(1.31 - 0.04)$). Lastly, the same offset for GPT-4 is 1.16 ($= -(-1.12 - 0.04)$) ($SE = 0.05$) with a corresponding probability of 0.92 ($= \sigma(1.31 + 1.16)$). The slopes of each log odds curve can then be calculated from the token count and token count interaction effects, as can be seen in Table 1.

Token Count	-2σ (164)	0σ (346)	2σ (529)	4σ (711)	6σ (893)
Davinci-002	0.87	0.55	0.18	0.04	0.01
Davinci-003	0.78	0.78	0.78	0.77	0.77
GPT-4	0.91	0.92	0.93	0.94	0.95
Grand Mean	0.79	0.79	0.78	0.78	0.77

Table 1: Probabilities of each model correctly answering an MCQ of a particular length.

If we assume that the hypothesis space of the logistic function can adequately describe the change in probability, then these results indicate that Davinci-002's accuracy rapidly degrades with increased token count, swinging from 0.87 down to 0.01 across the full range of MCQ token counts (164 to 893 tokens).[2] The curve for Davinci-003 also shows a negative relationship, though it is much more slight and can be seen to be in closer agreement with the corresponding LOESS plot. Lastly, GPT-4's curve clearly displays a slight positive relationship as well as a much higher overall predicted accuracy in the low 0.90s. Regardless of whether the hypothesis space is valid, it seems reasonable to infer the general trend of divergence in LM accuracy with respect to increasing token count, as both the LOESS and logistic mixed effects models agree on this point.

Looking at the second regression table, we may repeat the same calculations for the syntactic density relation and arrive at the probabilities in Ta-

[1]As such, we may ignore the p-values listed in the `summary` function.

[2]Though the stark disagreement between this logistic curve and the non-monotonic LOESS spline should caution us against simple interpretation.

Syntactic Density	-2σ (3.15)	0σ (3.40)	2σ (3.65)	4σ (3.90)
Davinci-002	0.51	0.55	0.59	0.63
Davinci-003	0.75	0.78	0.81	0.84
GPT-4	0.91	0.92	0.93	0.94
Grand Mean	0.76	0.79	0.79	0.8

Table 2: Probabilities of each model correctly answering an MCQ of a particular syntactic density (i.e. nodes per token).

ble 2. Note that the 0σ column is identical to the one in Table 1 when rounded to the hundredths place, meaning that the fixed effects intercepts are the same between models, while the slopes differ between models. In addition, the log odds curves for the syntactic complexity model are almost parallel, which reflects the lack of statistical significance of the `scaled_con:LLM` term.

4.2 Random Effects Interpretation

Our models represent the random effects as random intercepts by MCQ cluster. And while we attempted to fit a model with both random intercepts and slopes, there was not enough variation between MCQs within clusters to fit the random slopes.[3] As such, we were only able to characterize the variation in performance *between* MCQ clusters. This motivated us to identify MCQ properties that are highly *discriminative* between LMs in a post hoc analysis.

First, we extracted the 100 "easiest" and 100 "hardest" clusters from the random effects structure of the models. These data show that for both models, the easiest cluster is `high17038.txt`, and the hardest cluster is `high22834.txt`. Both models agree that the MCQs in the easiest cluster have an 0.91 probability of being answered correctly, and the MCQs in the hardest cluster have a corresponding probability of 0.36[4], which indicates that the clusters in RACE-h offer a wide range of difficulty levels for the mean LM. We visualize the full mixed effects structure in Appendix B.

5 Post Hoc Analysis

In our final experiment, we took the two lists of easiest and hardest MCQ clusters and compared the relative frequencies with which different *wh*-

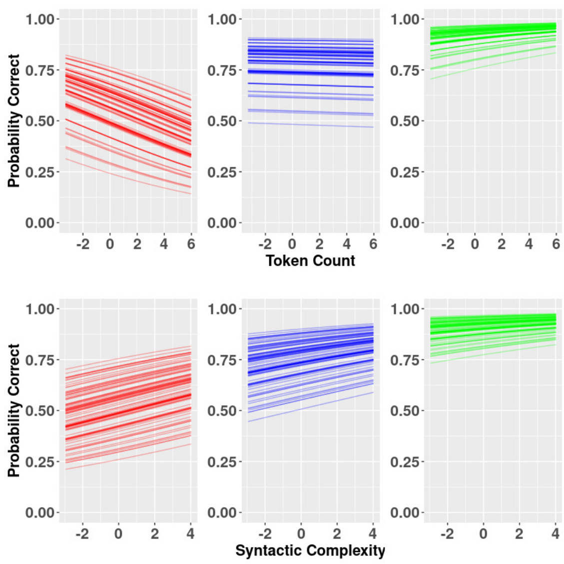

Figure 3: 100 randomly sampled random effects for Davinci-002, Davinci-003, and GPT-4

words occur as the first token in the interrogative portion of *each individual* MCQ. We obtained the contingency table that can bee seen in Table 3 upon which Fisher's exact test with a simulated p-value showed a significant difference in distribution between the easy and hard groups ($p = 0.01$, 2000 replicates). This result may indicate that "What", "Who", and "How" questions are on average more difficult for LMs to answer, while "Which", "Why", and "When" questions tend to be easier.

We then take a closer look at MCQ clusters with high *discriminatory power* (P_d), which can be thought of as the reliability with which an MCQ from a particular cluster can be used to distinguish between multiple LMs based on the correctness of their answers. We define this measure based on the average pairwise statistical distance between distributions of MCQs answered correctly in each cluster by each LM, as follows:

$$P_d = \frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} \overline{D}_{KL}(P_i \| P_j) \quad (4)$$

$$\overline{D}_{KL}(P_i \| P_j) = \frac{D_{\mathrm{KL}}(P_i \| P_j) + D_{\mathrm{KL}}(P_j \| P_i)}{2} \quad (5)$$

where $D_{\mathrm{KL}}(P_i \| P_j)$ is the Kullback-Leibler divergence between P_i and $P_j \in \{P_1, ..., P_n\}$, and n is the number of LLMs being compared.[5] Also, note that $P_k \sim \mathrm{Bernoulli}(p_k)$ where $0 \leq p_k \leq 1$.

[3]This is likely due a lack of verisimilitude in the way we measured token counts and syntactic complexity on each MCQ, which did not account for the internal structure of MCQs.

[4]Recall that the grand mean probability for all MCQs is 0.79 across the dataset.

[5]We chose to define P_d using the arithmetic mean of multiple measurements of the KL divergence because of its asymmetry and because we wanted to capture something analogous to variance for probability distributions.

We chose to calculate the KL divergence using a smoothing constant $\epsilon = 10^{-10}$ and the natural logarithm as follows:

$$q_k = \max(\min(p_k, 1-\epsilon), \epsilon) \tag{6}$$

$$D_{\mathrm{KL}}(P_i \| P_j) = q_i \ln\left(\frac{q_i}{q_j}\right) + (1-q_i)\ln\left(\frac{1-q_i}{1-q_j}\right) \tag{7}$$

We then find the MCQ clusters with the 100 highest and 100 lowest P_d values to construct another $wh-$ word contingency table. Again, Fisher's exact test revealed a significant difference in distribution between the high and low P_d groups ($p = 0.02$, 2000 replicates). We observe that while "When", "Which", "Where", and especially "Why" questions tend to have low P_d, "What", "Who", and especially "How" questions tend to have high P_d. Because these question types are both highly discriminative and tend to be harder than others, we may infer the possibility of a positive relationship between MCQ difficulty and and discriminative power. We also observe that "When", "Which", and "Why" questions are both easier and have lower P_d on average, reinforcing the plausibility of this relationship.

We also speculated that looking at the discriminatory power of a dataset item could be used to identify MCQs that have outsized usefulness in future LM benchmarking tasks. The distribution of these P_d values can be seen in Figure 4 and inform our final post hoc experiements. In part, we sought to determine if the P_d could be used to predict ill-formed MCQs that should be removed from the dataset for being unanswerable. We reasoned that low P_d may indicate a serious issue with the structure or logic of an MCQ, which would cause every LM to either miss the correct answer or even for every LM to get the answer right, because it does not depend on the contextual passage. To accomplish this, we split the dataset into four intervals based on the P_d value of each MCQ ranging from 0 to 15.4 and manually labeled 16 randomly sampled MCQs from each interval, obtaining the contingency table seen in Table 4. Fisher's exact test did not reveal a significant difference between the proportions of ill-formed MCQs between intervals ($p = 1$, 2000 replicates). This failure to reject the null hypothesis may indicate either that P_d is not useful for detecting dataset errors or that the sample size of 16 MCQs per interval does not provide sufficient statistical power to detect such an effect.

Lastly, we wanted to see whether a subset of the data based on a minimum P_d cutoff value could be used to reliably rank LMs by their task level accuracy without having to use the entire dataset. This experiment was theoretically motivated by the observation that not all MCQs are equally useful for distinguishing between different LMs, as some provide more information about the subject than others given a known population of LMs. To answer this question, we first filtered the dataset by the P_d values of each MCQ, discarding MCQs below four different threshold levels: 0, 5, 10, and 15. We then calculated the accuracy of each LM on each filtered subset to arrive at the data in Table 5 which confirm the statistical validity of the P_d construct, suggesting that a dataset's P_d scores may be useful even when ranking LMs outside of the original sample used to calculate those scores. In other words, we see the potential usefulness of P_d filtering in transfer evaluation due to its ability to emphasize the accuracy difference between different LMs even when restricted by relatively small sample sizes.[6]

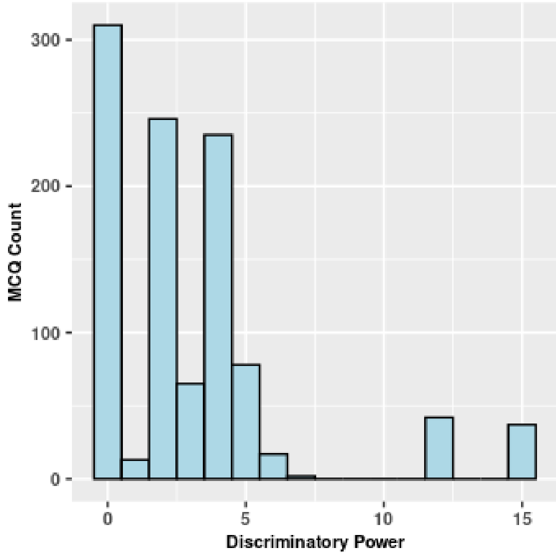

Figure 4: Histogram showing the distribution of P_d values across the 1,045 MCQs in the dataset ranging from 0 to 15.351 with mean P_d of 3.131.

6 Conclusion

6.1 Results

We have used interpretable statistical techniques to aid in the discovery of linguistic features that affect the difficulty that LMs have in multiple choice read-

[6]As long as the sample size is at least 30, the standard error of the accuracy measure can be no greater than $\sqrt{1/120} \approx 0.09$.

	What	When	Which	Why	Where	Who	Whom	How	Other
Easiest	62	9	58	21	3	1	1	7	252
Hardest	86	5	46	11	3	5	0	14	201
E/H	0.72	1.80	1.26	1.91	1.00	0.20	N/A	0.50	1.25
Highest P_d	63	5	39	4	3	2	0	13	158
Lowest P_d	52	12	47	15	4	1	0	5	165
H/L	1.21	0.42	0.83	0.27	0.75	2.00	N/A	2.60	0.96

Table 3: Wh- word contingency table for MCQs from the 100 easiest and 100 hardest MCQ clusters as well as the 100 most and 100 least discriminative clusters. Because clusters have a variable number of MCQs, the MCQ row totals also vary.

P_d	$[0, 5)$	$[5, 10)$	$[10, 15)$	≥ 15
Valid	13	12	12	12
Invalid	3	4	4	4

Table 4: Contingency table for all 1,045 MCQs split into four intervals of measured P_d.

$P_d >$	0	5	10	15
Davinci-002	0.54	0.19	0.06	0.11
Davinci-003	0.76	0.71	0.63	0.83
GPT-4	0.90	0.97	0.98	0.96
Grand Mean	0.73	0.62	0.56	0.63
MCQ Count	1,045	154	79	37
Question Count	3,498	449	202	76

Table 5: Task accuracy for each LM at each P_d cutoff level. It can be seen from the data that the accuracy of each LM diverges as the dataset is winnowed down.

ing comprehension tasks. Instead of using the typical task-level benchmarking evaluation methods, we fit two GLMER models to examine the fixed and random effects structures of LM performance on the RACE-h dataset to obtain a more granular view of how performance varies with respect to the features of particular dataset items. These findings then motivated an exploratory post hoc analysis where we compared the relative frequencies of different leading position $wh-$ words in different sub groups of MCQ clusters to determine the lexical properties that affect performance.

Our results demonstrate that performance diverges as MCQ token count grows, that it uniformly increases in log odds as syntactic complexity grows, and that there exists a positive relationship between the difficulty of an MCQ and its discriminative power. Each of these conclusions offers meaningful insights into how we can improve LM evaluation methodologies and explain LM behavior in response to different types of natural language prompts. Of particular interest is the surprisingly positive effect of syntactic complexity on the likelihood of comprehension, which runs counter to the standard assumptions made in the psycho-linguistics literature (Eslami, 2014) and invites deeper inquiry.

6.2 Discussion

Our experimental results have significant implications for how LM evaluation methodologies can be made more effective, efficient, and interpretable in various ways. The methods we used can be extended to aid in the discovery and explanation of additional linguistic features that predict LM performance on a variety of common tasks similar to MCQA MRC. This improved understanding would make it possible to construct more linguistically informed evaluation datasets that leverage feature engineered MCQs to more accurately and meaningfully compare the behavior of different LMs under varying contextual and meta-parameter settings at scale. Such an approach could yield a great bounty of practical and theoretical insights into the nature of LM behavior and their natural language understanding ability.

We have several recommendations for future lines of inquiry that could leverage our empirical and methodological results to great effect. For instance, datasets could be filtered for MCQs with high discriminatory power, which would yield smaller datasets that could be used to more efficiently rank LMs by performance. In addition, it would be beneficial to search for similar metrics that could leverage the observed performance of relatively few LMs to flag potential garbage data. This would be of very helpful for the NLP research and engineering communities, as many commonly used datasets for benchmarking performance have significant quality control issues. Indeed, having a set of statistical and linguistic tests that can be used to screen and filter datasets to ensure consistently high quality would be very useful to practitioners of all kinds.

7 Limitations

While we have striven to provide rigorous empirical justification for our conclusions and avoid data dredging, our experimental design still has some shortcomings that must be addressed. One issue that may limit the generalizability of our findings is that we only used three proprietary LMs to answer the question of how LM behavior is affected by different linguistic features. As such, the inferred relationships may not hold for out of distribution LMs. Another limitation of our methodology was that we measured linguistic features across entire dataset items rather than taking piece-wise measurements by passage, by question, and by answer. It was only after fitting the GLMER models that we realized piece-wise feature measurement would have offered many advantages to model interpretation, as it would have revealed additional variance within MCQ clusters, allowing us to fit a random slopes model as well.

Our method of measuring the token count was also less than ideal, as we used the NLTK library (Xue, 2011) to compute this value for each MCQ rather than simply count the number of leaf nodes provided in the Stanza parse trees. This way both the token count and the constituency complexity would have been derived from the same data structure and the same parsing algorithm.

Lastly, our results may not generalize to languages other than English, as the RACE-h dataset is a monolingual English dataset, and the observed distribution of syntactic density scores is not likely to hold for other languages.

8 Ethics Statement

We were assisted in the process of data labeling for the purpose of error estimation by a former graduate student trained in computational linguistics whose native language is Chinese but completed their education in the United States. We paid them \$20 per hour for a total of four hours to carefully label 32 MCQ clusters as either valid or invalid. The remaining 32 clusters were labeled by one of the authors whose native language is American English.

Aside from using the OpenAI API for model evaluation, we used the ChatGPT web interface to assist in information retrieval to help the us choose the best statistical techniques to answer our research questions. In addition, we used the web interface to create quick mock ups of figures, tables, and diagrams in LaTeXand TikZ, which we modified as needed to accurately represent our methods and findings. All assistance from ChatGPT was confined to the web interface and kept separate from the API calls so as to avoid any bias introduced via cross contamination of data.

Lastly, we do not foresee any negative societal repercussions from having run these experiments or from publishing our findings, as we have not trained any LMs. Rather, we have only used a small handful of them for a limited number of inferences, which is a figurative "drop in the bucket" with respect to climatological concerns.

References

R.H. Baayen, D.J. Davidson, and D.M. Bates. 2008. Mixed-effects modeling with crossed random effects for subjects and items. *Journal of Memory and Language*, 59(4):390–412. Special Issue: Emerging Data Analysis.

Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. 2015. Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(1):1–48.

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 610–623.

Taylor Berg-Kirkpatrick, David Burkett, and Dan Klein. 2012. An empirical investigation of statistical significance in nlp. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 995–1005.

Samuel R. Bowman and George E. Dahl. 2021. What will it take to fix benchmarking in natural language understanding? *CoRR*, abs/2104.02145.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *CoRR*, abs/2005.14165.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Hedayat Eslami. 2014. The effect of syntactic simplicity and complexity on the readability of the text. *Journal of Language Teaching and Research*, 5(5):1185.

Dustin Fife. 2023. *flexplot: Graphically Based Data Analysis Using 'flexplot'*. R package version 0.19.1.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *CoRR*, abs/2009.03300.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*.

Timothy Niven and Hung-Yu Kao. 2019. Probing neural network comprehension of natural language arguments. *arXiv preprint arXiv:1907.07355*.

OpenAI. 2023. Gpt-4 technical report.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

R Core Team. 2021. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Stefan Riezler and Michael Hagmann. 2022. *Validity, Reliability, and Significance: Empirical Methods for NLP and Data Science*. Synthesis Lectures on Human Language Technologies. Springer Cham.

Ananya B. Sai, Akash Kumar Mohankumar, and Mitesh M. Khapra. 2020. A survey of evaluation metrics used for NLG systems. *CoRR*, abs/2008.12009.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

B.T. West, K.B. Welch, and A.T. Galecki. 2022. *Linear Mixed Models: A Practical Guide Using Statistical Software*. CRC Press.

Nianwen Xue. 2011. Steven bird, evan klein and edward loper. natural language processing with python. o'reilly media, inc.2009. isbn: 978-0-596-51649-9. *Natural Language Engineering*, 17(3):419–424.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 93–104, Brussels, Belgium. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Rowan Zellers, Ari Holtzman, Elizabeth Clark, Lianhui Qin, Ali Farhadi, and Yejin Choi. 2020. Evaluating machines by their real-world language use. *CoRR*, abs/2004.03607.

Mariia Zyrianova, Dmytro Kalpakchi, and Johan Boye. 2023. Embrace: Evaluation and modifications for boosting race. *arXiv preprint arXiv:2305.08433*.

A Syntactic Measurement Examples

Here we provide examples of our measurement procedure for some toy sentences. The actual procedure we used when processing the data set differs only slightly in that the node count is summed over all of the sentences in an MCQ before dividing by the MCQ's token count. However, one oversight in our methodology was to use the NLTK library (Xue, 2011) to count the tokens in an MCQ, rather than simply counting the total leaf nodes in each of the trees generated by Stanza.

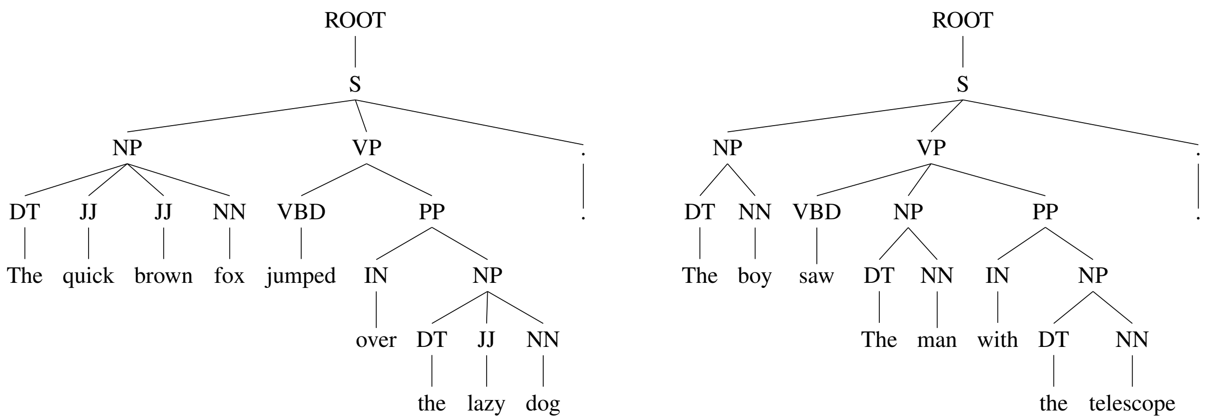

Figure 5: Constituency parse trees for "The quick brown fox jumped over the lazy dog." and "The boy saw the man with the telescope." The first sentence has 26 nodes in total, and 10 are leaf nodes that represent tokens. Thus, the raw syntactic complexity of this sentence would be 2.60 $(= 26/10)$. Accordingly, the second sentence would have a raw score of 2.78 $(= 25/9)$.

B R Formulas and Code

B.1 GLMER for Token Count

```
> intercepts_tok <- glmer(correct ~ scaled_tok * LLM + (1|passage), data=df, family=binomial)
> summary(intercepts_tok)
Generalized linear mixed model fit by maximum likelihood (Laplace Approximation) ['glmerMod']
 Family: binomial  ( logit )
Formula: correct ~ scaled_tok * LLM + (1 | passage)
   Data: df

     AIC      BIC   logLik deviance df.resid
 10675.3  10726.1  -5330.7  10661.3    10487

Scaled residuals:
    Min      1Q  Median      3Q     Max
-4.8964 -0.6874  0.3349  0.5810  1.8478

Random effects:
 Groups  Name        Variance Std.Dev.
 passage (Intercept) 0.5564   0.7459
Number of obs: 10494, groups:  passage, 1045

Fixed effects:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)      1.306962   0.037221  35.113  < 2e-16 ***
scaled_tok      -0.012971   0.032554  -0.398  0.69030
LLM1            -1.116906   0.035076 -31.843  < 2e-16 ***
LLM2            -0.044386   0.035716  -1.243  0.21396
scaled_tok:LLM1 -0.097136   0.034660  -2.803  0.00507 **
scaled_tok:LLM2  0.003958   0.035950   0.110  0.91233
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
            (Intr) scld_t LLM1   LLM2   s_:LLM1
scaled_tok   0.017
LLM1        -0.288 -0.020
LLM2        -0.109 -0.021 -0.229
scld_t:LLM1 -0.020 -0.278  0.019  0.021
scld_t:LLM2 -0.016 -0.164  0.019  0.011 -0.199
```

Figure 6: R formula for and summary of the intercepts only, token count GLMER model. Note that no warnings were raised when fitting the model with the call to `glmer`.

```
> vif(intercepts_tok)
                   GVIF Df GVIF^(1/(2*Df))
scaled_tok     1.145968  1        1.070499
LLM            1.002362  2        1.000590
scaled_tok:LLM 1.147014  2        1.034885
```

Figure 7: Variance inflation factors for the intercepts only, token count GLMER model. The VIF values for each covariate indicate minimal correlation with other covariates.

B.2 GLMER for Syntactic Complexity

```
> intercepts_con <- glmer(correct ~ scaled_con * LLM + (1|passage), data=df, family=binomial)
> summary(intercepts_con)
Generalized linear mixed model fit by maximum likelihood (Laplace Approximation) ['glmerMod']
 Family: binomial  ( logit )
Formula: correct ~ scaled_con * LLM + (1 | passage)
   Data: df

     AIC      BIC   logLik deviance df.resid
 10677.3  10728.1  -5331.6  10663.3    10487

Scaled residuals:
    Min      1Q  Median      3Q     Max
-4.8474 -0.6859  0.3335  0.5802  1.8448

Random effects:
 Groups  Name        Variance Std.Dev.
 passage (Intercept) 0.5569   0.7463
Number of obs: 10494, groups:  passage, 1045

Fixed effects:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)      1.3075563  0.0372347  35.117  < 2e-16 ***
scaled_con       0.0832204  0.0318668   2.612  0.00901 **
LLM1            -1.1176523  0.0350764 -31.863  < 2e-16 ***
LLM2            -0.0434572  0.0357304  -1.216  0.22389
scaled_con:LLM1  0.0072892  0.0340999   0.214  0.83073
scaled_con:LLM2 -0.0003695  0.0358214  -0.010  0.99177
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation of Fixed Effects:
            (Intr) scld_c LLM1   LLM2   s_:LLM1
scaled_con   0.033
LLM1        -0.288 -0.029
LLM2        -0.108 -0.007 -0.230
scld_c:LLM1 -0.016 -0.269  0.018  0.007
scld_c:LLM2 -0.007 -0.133  0.006  0.034 -0.231
```

Figure 8: R formula for and summary of the intercepts only, syntactic complexity GLMER model. Note that no warnings were raised when fitting the model with the call to `glmer`.

```
> vif(intercepts_con)
                   GVIF Df GVIF^(1/(2*Df))
scaled_con     1.127094  1        1.061647
LLM            1.002758  2        1.000689
scaled_con:LLM 1.128399  2        1.030660
```

Figure 9: Variance inflation factors for the intercepts only, syntactic complexity GLMER model. The VIF values for each covariate indicate minimal correlation with other covariates.