When Word Order Matters: Unraveling Redundancy and Sequence Sensitivity in Language Models through Information Theory

Xuanda Chen[1,2], Siva Reddy[1,2,3], Timothy J. O'Donnell[1,2,4]

[1]McGill University, [2]Mila Quebec AI Institute,
[3]Facebook CIFAR AI Chair, [4]Canada CIFAR AI Chair
xuanda.chen@mail.mcgill.ca, siva.reddy@mila.quebec, timothy.odonnell@mcgill.ca

Abstract

Language models (LMs) often seem insensitive to word order changes in natural language understanding (NLU) tasks. We propose that this insensitivity is due to linguistic redundancy, where word order and other cues, like case markers, provide overlapping, redundant information. Our hypothesis is that LMs show reduced sensitivity to word order when it conveys redundant information, with the extent of this insensitivity varying across tasks. We quantify the informativeness of word order using mutual information (MI) between unscrambled and scrambled sentences. Our findings reveal that as word order becomes less informative, model predictions become more consistent between the original and scrambled sentences. This effect, however, varies by task. For instance, in SST-2, language models maintain prediction consistency even when Pointwise-MI (PMI) changes, whereas in tasks like RTE, consistency drops to near random as PMI decreases, highlighting the critical role of word order.

1 Introduction

Language is fundamental to human communication, and its structure is inherently compositional. Compositionality enables us to derive meaning from unseen sentences by combining the meanings of individual components. For example, in the sentence *The cat chased the mouse*, the overall meaning is constructed from the words *cat*, *chased*, and *mouse*, along with their specific order. Word order is crucial for sentence comprehension, as altering it can completely change the meaning. For instance, reversing the order to *The mouse chased the cat* conveys the opposite meaning.

Despite the importance of word order in human language processing, recent studies show that LMs are insensitive to it (Sinha et al., 2021a,b; Pham et al., 2021). For example, in natural language inference (NLI), where the task is to determine if a 'hypothesis' logically follows, contradicts, or is neutral with respect to a 'premise,' LMs can identify entailment even when word order is scrambled. Sinha et al. (2021a,b) found that LMs correctly infer relationships even with disrupted word order, such as between the premise *A soccer game with multiple males playing* and the hypothesis *Some men are playing a sport*, or when scrambled as *males playing soccer game A with multiple* and *playing a are sport Some men*.

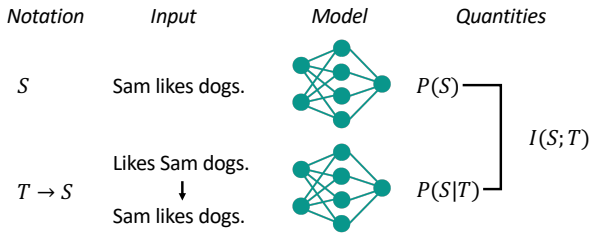

Figure 1: Variational approximation of the MI between scrambled and unscrambled sentences, using an LM and a reordering model. The estimation relies on bounding MI —see discussion in §2.

These findings question whether word order is essential for LMs to compute meaning. While altering word order in *The cat chased the mouse* changes its meaning, the sentence *The cat is chasing two balls* retains its meaning even when scrambled, e.g., *Is chasing two balls the cat*. This is due to subject-verb agreement and the animate subject requirement of *chase*. Thus, in such cases, redundant information from word order and grammatical agreement suggests that word order may not always be crucial for understanding.

In this paper, we introduce the *redundancy effect*, proposing that LMs may not rely on word order in NLU tasks because it provides redundant information. To measure redundancy, we treat scrambled and unscrambled sentences as random variables and use mutual information (MI) to measure the average amount of information scrambled sentences contain about unscrambled sentences

(Cover, 1999). This approach is independent of representation and processing, such as LM training or scrambling methods. We hypothesize that high MI indicates that word order is less critical for LMs in NLU tasks. Since MI is difficult to measure directly, we estimate it using a variational approximation with a reordering model (RM) and an LM (Li and Eisner, 2019).

Our findings show that LMs struggle with certain scrambled sentences with low Pointwise-MI (PMI) but can handle most NLU tasks with sentences that are easier to reorder. The redundancy effect varies by task, reflecting different sensitivities to word order. For tasks like SST-2, word order is largely irrelevant; predictions remain consistent despite changes in PMI. In contrast, for tasks like RTE, prediction consistency depends on PMI, with lower PMI sentences leading to less consistent performance.

2 MI Estimation

Mutual information (MI) measures the statistical dependence between random variables, such as between scrambled and unscrambled sentences. Let S be a random variable over possible sentences as strings of words, with distribution $P(S)$. Let T denote a random variable over all possible scramblings of S, and $\sigma(\cdot)$ be the scrambling function, such that $p(t) = \sum_{s \in \sigma(t)} p(s) \frac{1}{|s|!}$. The MI between unscrambled and scrambled sentences can be denoted as $I(S;T) \coloneqq \mathop{\mathbb{E}}_{s,t \sim p(s,t)} \left[\log \frac{p(s|t)}{p(s)} \right]$. This represents the expected logarithmic ratio of the probability of sentences given scrambling to the probability of sentences for all possible sentence and scrambling pairs. Estimating MI is challenging due to limited access to samples rather than the underlying distributions. To approximate MI, we replace the intractable conditional distribution $p(s|t)$ with a tractable variational distribution $q_\phi(s|t)$, providing a lower bound on MI due to the non-negativity of KL divergence.

$$I(S;T) = \mathop{\mathbb{E}}_{s \sim p(s)t|s \sim p(t|s)} \left[\log \frac{q_\phi(s|t)}{p(s)} \right] + \mathop{\mathbb{E}}_{t \sim p(t)} \left[KL(p(s|t)||q_\phi(s|t)) \right] \quad (1)$$

Given the fixed MI value and the non-negativity of KL divergence, maximizing the estimate is equivalent to minimizing KL by finding a q_ϕ that closely approximates the true distribution p. We define q_ϕ as the reordering model (RM), which aims to restore the original word order from scrambled sentences. The RM is trained on sentence-scrambling pairs (s, t) from a corpus, where each sentence corresponds to multiple possible scramblings. The optimal q_ϕ minimizes KL divergence and maximizes the expected value of $\log q_\phi(s|t)$ across all (s, t) pairs, effectively aligning q_ϕ with $p(s|t)$. The lower bound on MI serves as an approximation for the PMI of each sentence, calculated as $\text{pmi}(s;t) = \log_2 \frac{q_\phi(s|t)}{p(s)}$. Here $q_\phi(s|t)$ is estimated by the RM, and $p(s)$ is provided by a pre-trained LM, such as T5 (Raffel et al., 2020).

3 Training the Reordering Model

We trained the RM using a repurposed T5 model to estimate $q_\phi(s|t)$. The dataset comprised 100,000 sentences from English Wikipedia, each scrambled at the unigram level with six random seeds, yielding 600,000 (s, t) pairs. This was split into 90% for training and 10% for validation. The RM was trained for 10 epochs with a learning rate of 1e-4, taking about 5.6 hours on a single GPU.

4 Validating the Reordering Model

We further evaluated the RM's ability to infer original word order using additional linguistic cues like grammatical agreement and animacy. We tested the RM on a novel dataset not used during training, comprising two sentence types: type A and type B. Type A sentences, such as "Sam throws the rock," allow inference of subject and object through grammatical agreement or animacy. In contrast, type B sentences, like "Sam beats John," rely on word order for meaning, as grammatical agreement and animacy alone are insufficient for identifying the subject and object.

We hypothesize that the RM should excel at reconstructing the original order for type A sentences due to these cues, while its performance on type B sentences, where word order is critical, should be lower. We created two datasets of 1,000 sentences each using a context-free grammar and scrambled them with six random seeds. The RM's average accuracy was 0.948 for type A sentences and 0.506 for type B sentences. These results indicate that the RM effectively utilizes other linguistic cues to reconstruct word order when necessary.

5 Experiment, Data and Results

Recent studies have revealed that LMs are often insensitive to word order in NLU tasks. However, it is unclear if word order is always redundant across different contexts and tasks. According to the redundancy effect, LMs might disregard word order when it provides redundant information. To investigate this, we used a Bayesian mixed-effect logistic model to analyze the relationship between word order redundancy and LM performance in NLU tasks.

We quantified word order redundancy as the average PMI between an unscrambled sentence and all its possible scramblings. To approximate this, we scrambled each sentence with six random seeds and computed the average PMI from these samples. The response variable, consistency, indicated whether the LM produced the same label for both scrambled and unscrambled sentences, with 1 representing consistent predictions and 0 representing inconsistency. We also included sentence length as a confounding predictor to account for its effect on PMI estimation. Our model included both random intercepts and slopes unique to each task. Random intercepts captured task difficulty, with higher values indicating simpler tasks solvable by a bag-of-words approach. Random slopes reflected the redundancy effect specific to each task, with larger values indicating greater sensitivity to word order scrambling.

Consistent with Sinha et al. (2021a), we tested RoBERTa, a Masked LM proposed by Liu et al. (2019), and assessed its performance across a diverse range of binary classification tasks from popular benchmarks:

SST-2 (Socher et al., 2013) sentiment analysis task to classify one-sentence movie reviews as positive or negative.

MRPC (Dolan and Brockett, 2005) sentence similarity task to assess if pairs of sentences are semantically equivalent.

QQP (GLUE; Wang et al., 2019b) sentence similarity task to assess if a pair of Quora questions is semantically equivalent.

RTE (GLUE; Wang et al., 2019b) inference task to determine if a given hypothesis can be inferred from a premise as either 'entailed' or 'not entailed'.

COPA (Roemmele et al., 2011) inference task where a model selects the alternative most logically related to a premise sentence from two options.

BoolQ (Clark et al., 2019) QA task to answer multiple binary questions from a given passage.

WinoGrande (Sakaguchi et al., 2021) commonsense reasoning task to answer multiple-choice questions about typical event sequences based on common sense.

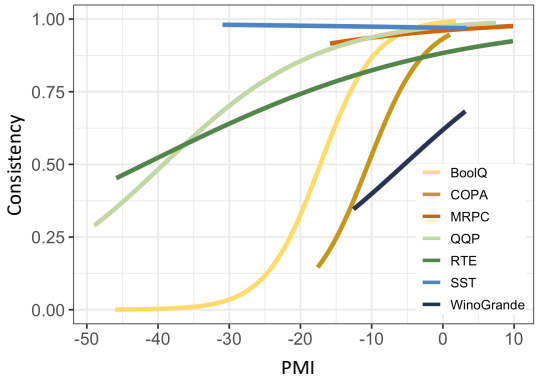

Figure 2: Curves illustrate the simulation of the linear model for the redundancy effect. The x-axis represents the data range, which varies by task. Line slopes indicate the degree of the redundancy effect, with steeper lines showing a more pronounced effect. The intercept reflects baseline PMI influence on Consistency and indicates task difficulty: lower intercepts denote tasks that are more challenging for LMs.

5.1 Regression Results

To assess the redundancy effect of word order on LMs in NLU tasks, we focused on test samples where RoBERTa's predictions were accurate, as inaccuracies could skew the analysis. Our model identified a significant redundancy effect at the group level. Specifically, PMI strongly predicted the consistency of RoBERTa's predictions ($\hat{\beta}$=1.87, SE=0.91, CI=[-0.09,3.67], ROPE>0.999). We further explored task variability by plotting the fitted curves for each task in Figure 2. The x-axis shows PMI (word order redundancy), and the y-axis represents consistency (LM performance). The curves revealed significant differences in intercepts and slopes across tasks. Tasks like SST-2 and MRPC displayed nearly flat curves, suggesting minimal reliance on word order. In contrast, BoolQ and COPA had steeper curves, indicating high sensitivity to word order scrambling, with performance dropping

sharply when word order was disrupted. Additionally, sentences in BoolQ, QQP, and RTE with low PMI were notably harder to reconstruct without the original word order. Overall, our findings indicate that while LMs can manage scrambled sentences when the original order is recoverable, sensitivity to word order varies by task.

5.2 Case Studies on Negative-PMI Sentences

We observed that many sentences have low negative PMI values, indicating that the LM finds the scrambled version of a sentence much less probable than the original. This suggests that the LM considers alternative word orders more favorable. Table 1 presents several manually checked examples of such reconstructed sentences. While some are grammatically correct, they can result in amusing or nonsensical meanings. For example, an original sentence about the impact of Donald Trump's presidency on India was reconstructed to ask about its effect on the U.S. if Trump became President of India. In another case, a sentence intended to convey a decrease in consumer numbers to 32 was reconstructed to suggest that consumers were committing suicide due to job scarcity. These examples also reveal the LM's strong bias toward declarative sentences; for instance, the LM mistakenly transformed a standard question into a declarative statement using the phrase 'why so.'

6 Related Work

Research has increasingly focused on LMs' insensitivity to word order scrambling. Initially, Sinha et al. (2021b) found that LMs retain high accuracy in natural language inference (NLI) tasks despite sentence scrambling. Similarly, Ettinger (2020) suggested that LMs might prioritize individual words over structural cues, explaining their insensitivity to word order. Sinha et al. (2021a) and Gupta et al. (2021) further explored this by training LMs on datasets with scrambled sentences and found no performance decline in NLI tasks. Pham et al. (2021) investigated this phenomenon across various NLU tasks from the GLUE (Wang et al., 2019b) and SuperGLUE (Wang et al., 2019a) benchmarks, noting that tasks like CoLA and RTE are more sensitive to word order scrambling. Clouatre et al. (2022) expanded on this by examining subword and character-level scrambling, finding that LMs can handle NLU tasks effectively as long as local structures remain intact. Papadimitriou et al. (2022) analyzed word order representation in LMs through grammatical role classification, revealing that word order is primarily processed in higher layers. In contrast, our study explores the variability in word order insensitivity across different sentences using mutual information. Our approach is agnostic to both the LM's representation and processing, independent of training schemes and scrambling strategies.

7 Discussion and Conclusion

This study explores when word order is crucial for LMs in natural language understanding (NLU) tasks. We introduce the concept of the redundancy effect, which suggests that word order may not be essential when it conveys redundant information that can be inferred from other linguistic cues. Our findings reveal that the redundancy effect varies across sentences, with some requiring the correct word order for comprehension. This effect also differs among NLU tasks, with some, such as COPA, being more sensitive to word order.

Additionally, we examine sentences with low negative PMI, demonstrating that these can be reordered in grammatically correct and semantically sensible ways. Our study aligns with recent findings on human interpretation mechanisms, which show that humans can often ignore certain word order errors due to their ability to derive meaning from other cues (Traxler, 2014; Mirault et al., 2018; Mollica et al., 2020). We propose that the redundancy effect may explain this phenomenon, as scrambled parts of word order may not provide additional information beyond what is already available from other linguistic signals.

In conclusion, we present computational evidence that linguistic redundancy can account for LMs' insensitivity to word order. Our regression model reveals a strong redundancy effect across all tasks, suggesting that LMs can handle scrambled texts when word order is less informative and recoverable. However, the effect varies: for tasks like SST-2, LMs' predictions remain consistent despite changes in PMI, while for tasks like RTE, consistency drops with lower PMI, indicating the crucial role of word order.

8 Limitations

Our analysis is limited to a narrow range of tasks and English-only data, which may overlook linguistic diversity, where the role of word order

Original Sentences	Generated Sentences
what would be the effect on India if Donald Trump really becomes the president of US?	if the president of the India becomes really Donald Trump what would be the effect on US?
how much time I need to wait for the result of South Korean student visa?	South Korean student wait for the result of visa how much time I need to?
consumers who said jobs are difficult to find jumped to 32	32 consumers who are said to be difficult to find jobs jumped
Germany is the powerhouse of the EU	the powerhouse of the EU is Germany
why were the Viking raids so successful?	the Viking raids were successful why so?

Table 1: Examples of probe model generation.

varies across languages. Expanding to typologically diverse languages would offer broader insights. While we used the T5 model, future work should explore and compare a wider range of models. Additionally, a more accurate mutual information estimation method could provide a clearer understanding of the relationship between word order and model consistency, offering deeper insights into redundancy effects in language models.

References

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. 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 2924–2936.

Louis Clouatre, Prasanna Parthasarathi, Amal Zouaq, and Sarath Chandar. 2022. Local Structure Matters Most: Perturbation Study in NLU. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3712–3731. Association for Computational Linguistics.

Thomas M Cover. 1999. *Elements of information theory*. John Wiley & Sons.

Bill Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Third International Workshop on Paraphrasing (IWP2005)*.

Allyson Ettinger. 2020. What bert is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Ashim Gupta, Giorgi Kvernadze, and Vivek Srikumar. 2021. Bert & family eat word salad: Experiments with text understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12946–12954.

Vladimir I Levenshtein et al. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union.

Xiang Lisa Li and Jason Eisner. 2019. Specializing word embeddings (for parsing) by information bottleneck. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Richard McElreath. 2020. *Statistical rethinking: A Bayesian course with examples in R and Stan*. Chapman and Hall/CRC.

Jonathan Mirault, Joshua Snell, and Jonathan Grainger. 2018. You that read wrong again! a transposed-word effect in grammaticality judgments. *Psychological Science*, 29(12):1922–1929.

Francis Mollica, Matthew Siegelman, Evgeniia Diachek, Steven T Piantadosi, Zachary Mineroff, Richard Futrell, Hope Kean, Peng Qian, and Evelina Fedorenko. 2020. Composition is the core driver of the language-selective network. *Neurobiology of Language*, 1(1):104–134.

Isabel Papadimitriou, Richard Futrell, and Kyle Mahowald. 2022. When classifying grammatical role, bert doesn't care about word order... except when it matters. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 636–643.

Thang Pham, Trung Bui, Long Mai, and Anh Nguyen. 2021. Out of order: How important is the sequential order of words in a sentence in natural language understanding tasks? In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1145–1160.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *AAAI spring symposium: logical formalizations of commonsense reasoning*, pages 90–95.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

Koustuv Sinha, Robin Jia, Dieuwke Hupkes, Joelle Pineau, Adina Williams, and Douwe Kiela. 2021a. Masked Language Modeling and the Distributional Hypothesis: Order Word Matters Pre-training for Little. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2888–2913.

Koustuv Sinha, Prasanna Parthasarathi, Joelle Pineau, and Adina Williams. 2021b. Unnatural Language Inference. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 7329–7346.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Matthew J Traxler. 2014. Trends in syntactic parsing: Anticipation, bayesian estimation, and good-enough parsing. *Trends in cognitive sciences*, 18(11):605–611.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019a. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019b. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019*.

Li Yujian and Liu Bo. 2007. A normalized levenshtein distance metric. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):1091–1095.

A Regression Model

A.1 Generalized Linear Model

Throughout our experiments, we utilize Bayesian logistic regression models. To begin, let us recall the definition of the classical linear regression model. Given a training set $\mathcal{D}$ consisting of inputs and targets $(\mathbf{x}^{(n)}, \mathbf{y}^{(n)})_{n=1}^{N}$, the regression model is defined as follows:

$$\mathbf{y} = \mathbf{w}^\top \mathbf{x} + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma_{err}^2) \tag{2}$$

Here, $\mathbf{y}$ represents the target variable, $\mathbf{x}$ represents the input variable, and $\mathbf{w}$ represents the learned weights that reflect the degree to which $\mathbf{y}$ is conditioned on $\mathbf{x}$. The error term ϵ represents the unexplained variance in $\mathbf{y}$. It is important to note that logistic regression differs from standard linear regression in the way that the probability of a particular outcome is linked to the linear predictor function. Specifically, the logit function, which is the natural logarithm of the odds, is used to convert the probability (which is bounded between 0 and 1) to a variable ranging over $(-\infty, +\infty)$. This transformation is used to match the range of the linear prediction function.

Maximum Likelihood Estimation (MLE) is often used to estimate the weight variable $\mathbf{w}$. The likelihood of the model is derived, and then maximized with respect to $\mathbf{w}$ using an optimization algorithm such as Gradient Descent. However, it is important to note that MLE assumes that the data is independently and identically distributed (i.i.d.). This assumption is not always satisfied, and the estimation can therefore be unreliable. To address this issue, we use Maximum A Posteriori (MAP) estimation, which is defined under the Bayesian framework and works on a posterior distribution (as opposed to the likelihood alone). The inclusion of the prior $P(\mathbf{w})$ in MAP leads to more robust estimation of parameters. In our case, we do not have balanced data points across different tasks, and some tasks have only a few samples. Bayesian regression helps us to make more robust comparisons between the estimated $\mathbf{w}$ across tasks by sampling from the actual shape of the posterior distribution and obtaining confidence intervals.

A.2 Generalized Mixed Effects

In our study, co-task samples are expected to exhibit dependencies in how the error terms are sampled. To model these dependencies through random effects, we adopt mixed-effects models. Specifically, for input vector $\mathbf{x}$ from a particular task k, our model is defined as follows:

$$\mathbf{y} = \mathbf{w}^\top \mathbf{x} + \mathbf{r}_k + \epsilon \tag{3}$$

Here, $\mathbf{y}$ denotes the target variable, $\mathbf{w}$ is the learned weight vector that captures the degree to which $\mathbf{y}$ is dependent on $\mathbf{x}$, and ϵ represents the unexplained variance in $\mathbf{y}$. In addition, the random

intercept $\mathbf{r}$ reflects the individual differences in the mean across all conditions and is assumed to be shared across all samples in task k.

To estimate individual differences in the effect of a predictor, i.e., different $\mathbf{w}$ for each task, we can add random slopes to the model. The updated formulation is as follows:

$$\mathbf{y} = \mathbf{w}^\top \mathbf{x} + \mathbf{s}k^\top \mathbf{x} + \mathbf{r}k + \epsilon \quad (4)$$

Here, each $\mathbf{s}k \sim \mathcal{N}(0, \Sigma\mathbf{s})$ represents a task-specific random slope, and $\Sigma_\mathbf{s}$ is a learned covariance matrix that shows the variance of $\mathbf{s}$ across tasks. We interpret random intercepts as the baselined task difficulty and random slopes as the sensitivity to word order.

B Response Variable and Predictors

B.1 consistency as response

We aim to investigate whether the LMs can accurately predict the meaning of scrambled texts, comparable to the predictions it makes on original texts. To this end, we define an *consistency* that measures the consistency between the LM's predictions on scrambled texts and the corresponding predictions on the original texts. If the predictions on scrambled texts and normal texts are consistent, we assign a value of 1 to the consistency; otherwise, we assign a value of 0.

To carry out this investigation, we utilize RoBERTa (Liu et al., 2019) as our prediction model and focus on test samples where it correctly predicts all the original texts. We then filter a new dataset, denoted as $\mathcal{D} = (x, y)|f(x) = y$, where $f(x)$ represents the LM's prediction on text input x. We use this dataset to train a regression model that models the consistency as a function of various predictors.

B.2 PMI and Sentence length as predictors

We aim to investigate the effect of point-wise mutual information (PMI) on the consistency of scrambled texts, with the goal of shedding light on the redundancy effect. As such, PMI is included as a predictor in our regression model. Additionally, it is important to consider sentence length as a potential confounder, as it may also influence the consistency. However, including both predictors in the model may result in collinearity and unreliable coefficient estimation.

To assess the relationship between sentence length and PMI, we conducted a correlation analysis, as shown in Figure 3. Our findings suggest that while there is some correlation between the two variables, it is not substantial.

To address the potential collinearity, we fitted two regression models, one with sentence length as a predictor and one without. Based on Bayes Factors, the model including sentence length was found to better fit the data, and thus we have chosen to report statistics based on this model.

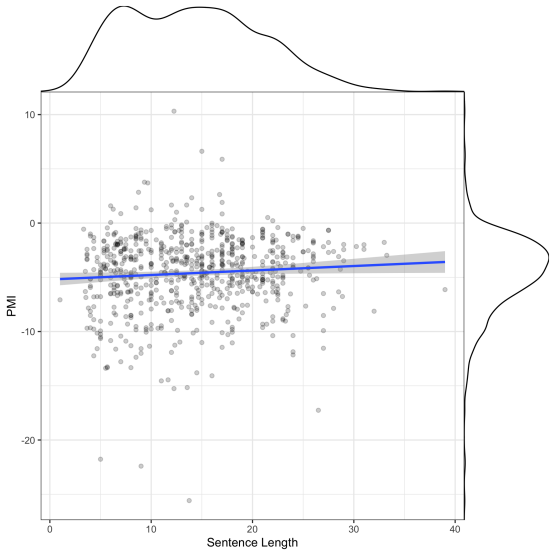

Figure 3: Correlation between sentence length and PMI.

B.3 Redundancy Effect at Group and Individual Level

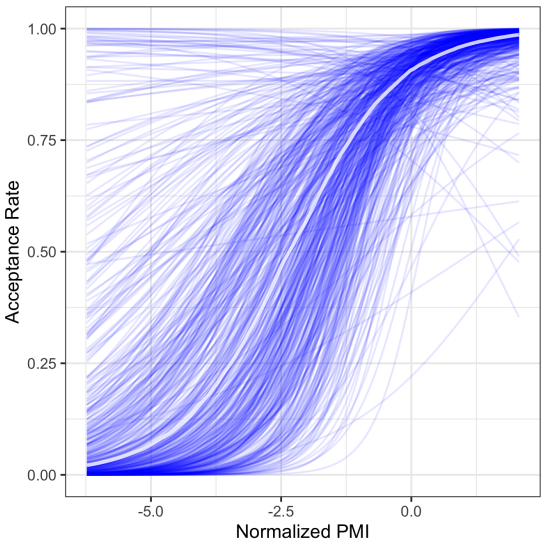

Figure 4: Conditional effect of PMI on consistency. The effect is estimated on random posterior draws from the model.

In Figure 5, we present the individual differences

in the mean across all conditions, represented by the random intercepts. A higher mean indicates a potential ceiling effect, whereby the baselined consistency remains high regardless of the sentence PMI. We sought to determine the statistical significance of these estimates, but Bayesian inference does not rely on statistical significance. Instead, we interpret the probability distribution, and values falling outside a predefined range are considered to have no practical effect or a negligible magnitude. This range is referred to as the region of practical equivalence (ROPE), and we set it at a range of -0.18 to 0.18 of a standardized parameter, which corresponds to a negligible effect size according to McElreath (2020). Our results indicate that all estimations, except for WinoGrande, are practically effective, indicating that all tasks, except WinoGrande, accept scrambled inputs significantly above chance.

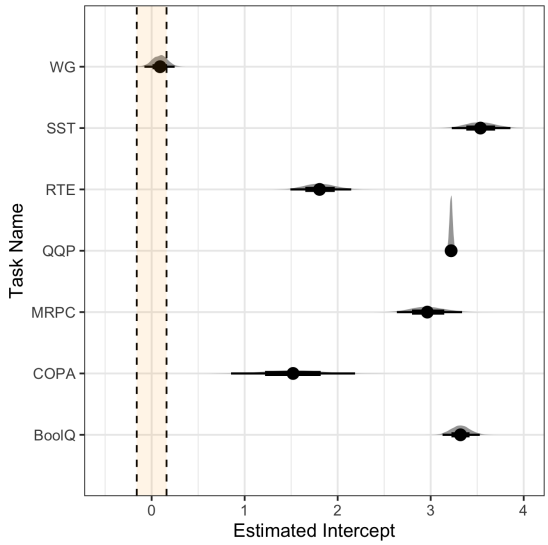

Figure 5: Estimated random intercepts across all tasks. The shaded area is the ROPE showing whether the effect size is big enough to be effective.

Figure 6 illustrates variations in the effect of PMI across tasks, which reflects the variance in sensitivity to word order. While SST-2 does not exhibit a notable effect, all other tasks show varying degrees of sensitivity, with BoolQ and COPA demonstrating the strongest.

C Reordering Model Evaluation

C.1 Probe accuracy on the CFG datasets

We endeavored to investigate the ability of a probe model to restore correct word order from perturbation, specifically focusing on two distinct types

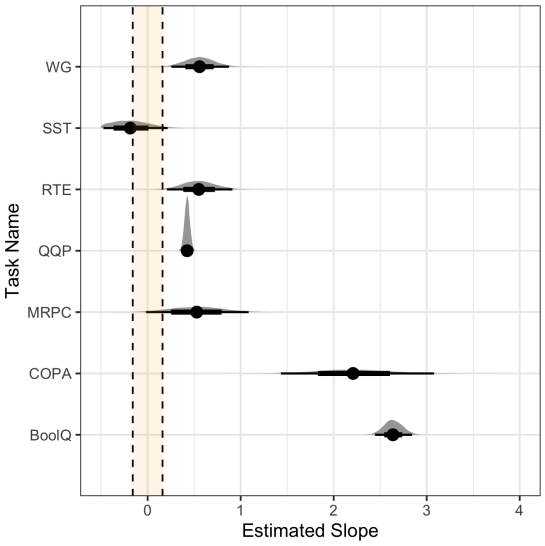

Figure 6: Estimated random slopes of PMI.

of argument structures - common noun and proper noun. To this end, we constructed two datasets using context-free grammar (CFG), each containing 1000 sentences with varying numbers of tokens, and subsequently scrambled them randomly six times, to evaluate the performance of the reordering model on a range of perturbations.

Our experimental results, presented in Figure 7, reveal that the reordering model exhibits high accuracy (0.948) in restoring common noun structures, while demonstrating chance performance (0.506) on the proper noun structures. Although the variance within proper noun accuracy is slightly higher than that within common noun accuracy, overall variance is low.

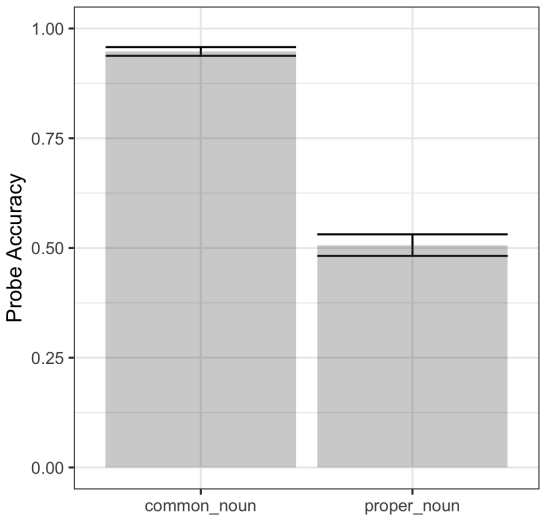

Figure 7: Probe accuracy on the CFG dataset.

C.2 Goodness of Probe Model

We aim to investigate the efficacy of the reordering model in restoring the original word order from perturbed sentences. To achieve this goal, we propose a metric based on the Levenshtein Distance (LD), which is a standard measure of the minimum number of single-character edits required to transform one sequence into another (Levenshtein et al., 1966; Yujian and Bo, 2007). Specifically, we define the Probe Accuracy (PA) as $1 - norm(LD(o, r))$, where o represents unscrambled sentences while r reconstructed sentences. To ensure methodological rigor, we include a control baseline that measures the perturbation degree. This Control Accuracy (CA) is computed as the same quantity between unscrambled and scrambled sentences.

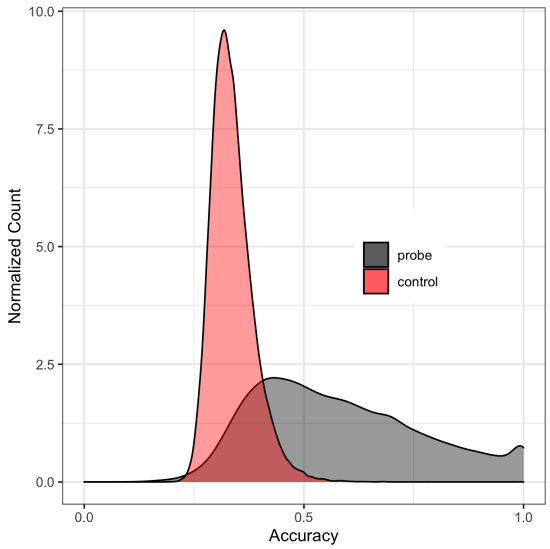

Figure 8: Distribution of PA and CA scores across tasks.

We present our findings on the PA and CA scores for all sentences in the tested tasks. As depicted in Figure 8, the CA scores exhibit a highly dense distribution around 0.3, indicating that scrambling may leave some local structures unscathed. However, the PA scores flatten the distribution of the CA scores, demonstrating that the reordering model can restore more local word order, although the reconstruction may not be perfect. An overlapping area between the PA and CA scores suggests that some sentences are difficult to recover, and the reordering model may struggle with such sentences. A more detailed presentation of the results can be found in Figure 9.

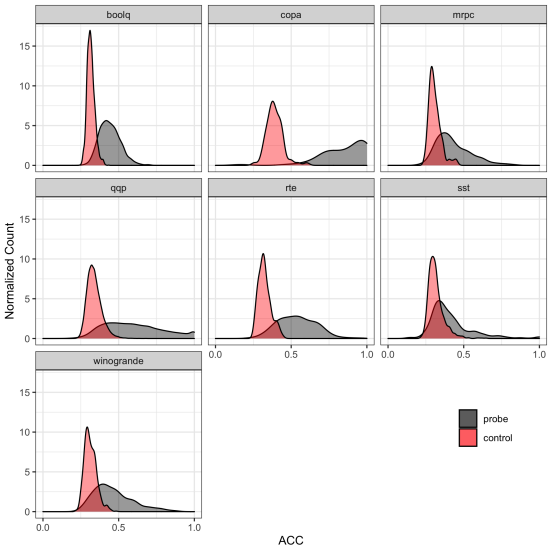

Figure 9: Distribution of sentence-level PA and CA scores across tasks.

D More information on datasets

We have curated a selection of natural language understanding (NLU) tasks from popular benchmarks for our study. Table 2 provides a comprehensive overview of these datasets (including the number of sentence pairs from train and validation datasets, the average sentence length, and the number of sentences), highlighting their differences in terms of size and complexity. While some are considerably large, such as QQP, others are relatively small, such as COPA.

Tasks	Samples		Sent Stat	
	Train	Val	#Len	#Num
SST-2	67349	872	16.912	915
MRPC	3668	408	17.512	1162
QQP	363846	40430	10.480	87635
RTE	2490	277	16.772	943
COPA	400	100	5.991	300
BoolQ	9427	3270	19.677	18614
WG	40398	1267	17.062	1515

Table 2: Statistics of selected NLU tasks (WG short for WinoGrande). Sentence statistics are calculated on the validation sets.

To gain a more in-depth understanding of the task design, we have included specific examples from each of these tasks, illustrating both the input and output components in Table 10.

We note that the difficulty of reconstructing word order from scrambling varies across tasks. Figure 10 showcases the distribution of by-task PMI

scores, where higher PMI scores generally imply that the original word order is easier to restore.

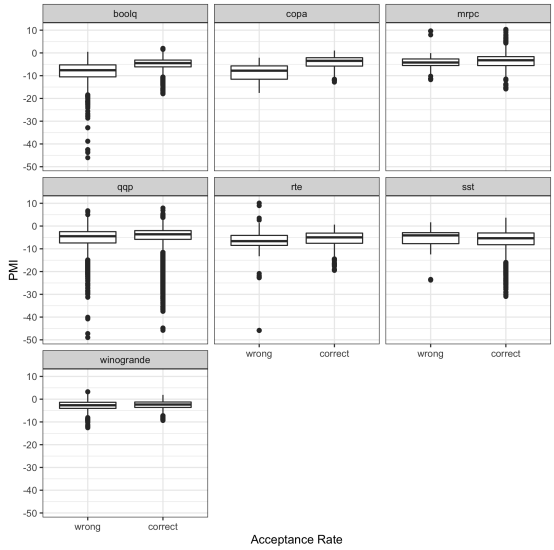

Figure 10: Boxplots of PMI distribution across tasks.