Linguistic Blind Spots of Large Language Models

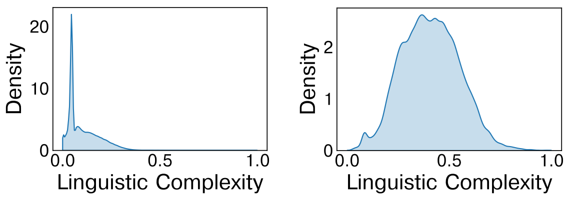
Jiali Cheng **Hadi Amiri**
University of Massachusetts Lowell
{jiali_cheng, hadi_amiri}@uml.edu

Abstract

Large language models (LLMs) are the foundation of many AI applications today. However, despite their remarkable proficiency in generating coherent text, questions linger regarding their ability to perform fine-grained linguistic annotation tasks, such as detecting nouns or verbs, or identifying more complex syntactic structures like clauses in input texts. These tasks require precise syntactic and semantic understanding of input text, and when LLMs underperform on specific linguistic structures, it raises concerns about their reliability for detailed linguistic analysis and whether their (even correct) outputs truly reflect an understanding of the inputs. In this paper, we empirically study the performance of recent LLMs on fine-grained linguistic annotation tasks. Through a series of experiments, we find that recent LLMs show limited efficacy in addressing linguistic queries and often struggle with linguistically complex inputs. We show that the most capable LLM (Llama3-70b) makes notable errors in detecting linguistic structures, such as misidentifying embedded clauses, failing to recognize verb phrases, and confusing complex nominals with clauses. Our results provide insights to inform future advancements in LLM design and development.

1 Introduction

Large Language Models (LLMs) have revolutionized NLP by achieving remarkable performance on a wide range of tasks and applications, including zero-shot inference (Weller et al., 2020; Brown et al., 2020); solving math problems (Wei et al., 2022); representing human emotions (Li et al., 2024); and serving as planners (Huang et al., 2022), conversational agents (Ouyang et al., 2022), or text-to-code convertors (Sun et al., 2023). Nevertheless, despite recent studies (Shen et al., 2021; Yu et al., 2023; Chen et al., 2024) aiming to understand Transformers (Vaswani et al., 2017) as the building block of LLMs, there is a lack of systematic evaluation of their ability in performing fine-grained linguistic annotation tasks.



(a) Penn Treebank (b) CoNLL 2000

Figure 1: Distribution of linguistic complexity in two widely-used NLP datasets. The plots show **(a)**: a strong skew toward linguistically simple examples in the Penn Treebank and **(b)**: a concentration around moderate complexity in CoNLL 2000, which highlights an over-representation of easier or medium-difficulty samples in the datasets.

Recent work studied LLMs from different linguistic perspectives, including grammar learning with small models (Huebner et al., 2021), effect of pre-training on learning linguistic properties like the depth of parse tree or verb tense (Alajrami and Aletras, 2022), the role of individual neurons in POS tagging and chunking tasks (Durrani et al., 2020), and the effect of prompt design for detecting linguistic properties (Blevins et al., 2023). However, existing evaluations are based on NLP datasets where linguistically "easy" examples (see Section 2) are overrepresented. For instance, Figure 1 shows histograms of the linguistic complexity of samples in two widely-used NLP datasets: Penn Treebank (Marcus et al., 1993) and CoNLL 2000 (Tjong Kim Sang and Buchholz, 2000). The skewed distribution toward linguistically easy or medium examples can artificially inflate performance on NLP tasks[1] and prevent true

[1]This phenomenon has challenged the NLP community across natural language inference (NLI), POS tagging, and parsing tasks, where models show human-level performance, while lacking cognitive ability to address these tasks. For example, recent work by Sinha et al. (2021) shows that BERT

Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics, pages 1–17

evaluation of models in NLP. We mitigate this bias by reducing the effect of overrepresented examples, i.e., categorizing samples based on their linguistic complexity and uniformly sampling data from distinct groups for a more reliable assessment.

We investigate the following research questions: (1): *how accurately can recent LLMs detect complex linguistic structures in input text?* (2): *which linguistic structures represent the blind spots of recent LLMs–meaning the most challenging for them?* (3): *how does the performance of LLMs vary across different levels of linguistic complexity of inputs?* We answer these questions by designing an empirical study for LLMs. The contributions of this paper are in examining recent LLMs's ability to detect specific linguistic structures across varying levels of linguistic complexity, providing meaningful insights into their limitations and biases, and highlighting potential avenues for future improvements.

Experimental results show that recent LLMs have limited efficacy in addressing linguistic queries, particularly struggling with complex linguistic structures such as complex nominals and T-units. In particular, Llama3-70b and GPT-3.5 are the most capable models among evaluated LLMs, while still making mistakes on simple linguistic queries. In addition, the performance of all evaluated LLMs often substantially fluctuates as sample complexity varies.

2 Background

Linguistic Complexity: quantifies the variability and sophistication in productive vocabulary, grammatical structures, and fluency in text data. It has been extensively investigated in psycholinguistics literature (Wolfe-Quintero et al., 1998; Zareva et al., 2005; Lu, 2010; Housen et al., 2019; Biber et al., 2020); and examined in quantifying language proficiency (Yannakoudakis et al., 2011; Lu, 2012), readability assessment and text simplification (Feng et al., 2009; Xu et al., 2015; Xia et al., 2016; Lee et al., 2021), and improving NLP tasks (Wei et al., 2021).

Lexical Complexity: is concerned with lexical *density*, *sophistication*, and *variation*. Lexical density is often quantified by the extent of information-carrying words in inputs. Lexical sophistication measures the proportion of *sophisticated* or infrequent words in texts. Lexical Variation refers to the diversity of vocabulary in text. Examples include type-token ratio (Templin, 1957) and its variations including *D-measure* (Malvern et al., 2004), which determines lexical variation of text by finding the curve that best fits the actual relationship between types and tokens in input text.

Syntactic Complexity: determines variability and sophistication in grammatical structures. A sentence like "*the mouse ate the cheese*" can be converted to its well-formed yet complex counterpart "*the mouse the cat the dog bit chased ate the cheese*," which forces readers to suspend their partial understanding of the sentence by encountering subordinate clauses that substantially increase the cognitive load of the sentence. Syntactic complexity measures the length of production units at the clausal, sentential, or T-unit levels; the amount of subordination, e.g. number of clauses per T-unit; the amount of coordination, e.g. number of coordinate phrases per clause or T-unit; and the range of surface and particular syntactic and morphological structures, e.g. frequency and variety of tensed forms (Wolfe-Quintero et al., 1998; Ortega, 2003).

Linguistic Knowledge of LLMs Blevins et al. (2023) designed *structured* prompting to assess the linguistic capabilities of LLMs. They provided each LLM with fully labeled demonstrations, and a query sentence and its partially tagged version. Each predicted label was appended to the partially tagged query along with the next word to iteratively tag the full query. They found that GPT-3.5 is robust to arbitrary label selections and ignores labels conflicting with its prior knowledge, indicating that the models can learn general linguistic knowledge during pre-training, rather than simply memorizing the data. Alajrami and Aletras (2022) empirically compared linguistically-motivated (e.g. masked language modeling (Devlin et al., 2019)) and non-linguistically motivated (e.g. masked first character prediction (Yamaguchi et al., 2021)) pre-training objectives for BERT on linguistic probing tasks (Linzen et al., 2016; Warstadt et al., 2020). They found the two objectives achieve similar performance. Clark et al. (2019) showed that attention heads in transformers attend to boundary tokens, positional offsets, and whole sentence; while Voita et al. (2019) showed that attention heads mainly handle positions, syntax, and rare words. Dur-

is invariant to random word order permutation in case of NLI, which can be attributed to the high prevalence of linguistically easy samples in NLI datasets (Elgaar and Amiri, 2023b).

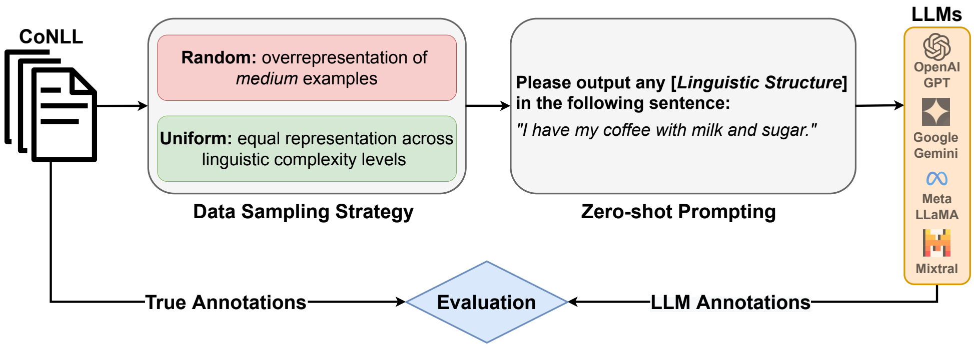


Figure 2: Workflow for finding linguistic blind spots of LLMs. As illustrated in Appendix A, GPT and other LLMs have good knowledge of our target tasks and the relevant terminology used in the prompts. [*Linguistic Structure*] in the prompts indicate any of the lexical or syntactic structures listed in Appendix C.

rani et al. (2020) compared linguistic knowledge learned by LMs at neuron level. They narrowed down neurons to a specific subset, located in lower hidden layers for lexical knowledge and in higher layers for semantic knowledge. Finally, Sharma et al. (2023) found that learning non-linguistic knowledge (e.g. numerical skills) sacrifices the linguistic knowledge of LLMs, and Ettinger (2020) found that BERT underperforms on commonsense, pragmatic inference, and negation tasks.

3 Finding Linguistic Blind Spots

We evaluate LLMs on recognizing specific linguistic structures (see below). For this purpose, we use gold linguistic annotations, lexical complexity analyzer from (Lu, 2012), and syntactic complexity analyzer from (Lu, 2012) to quantify linguistic complexity of samples. We note that the estimations provided by these tools have perfect agreement (based on Cohen's Kappa) with estimations provided by more recent linguistic complexity analysis tools (Lee et al., 2021; Lee and Lee, 2023).

Linguistic Structures: we consider different levels of granularity: **word-level structures** like nouns, verbs, adjectives, adverbs, prepositions, conjunctions, numerals, determiners, punctuation, particles, and words that cannot be assigned a part-of-speech (POS) tag; **phrase-level structures** like noun phrases (NP), verb phrases (VP), adjective phrases (ADJP), adverb phrases (ADVP), conjunction phrases (CONJP), complex nominals (CN); and **sentence-level structures** like clauses (C), dependent clauses (DT), T-units (T), and complex T-units (CT). Appendix C lists these structures.

Data Sampling Strategy The overrepresentation of easy and medium examples shown in Figure 1 suggests that the linguistic capability of LLMs may have been overestimated in existing literature (Blevins et al., 2023; Yang and Tu, 2022; Shen et al., 2018). For fair evaluation across the linguistic complexity spectrum, we divide samples into eight groups of increasing linguistic complexity, determined using (Lu, 2010, 2012), and uniformly at random sample from each group, leading to a total number of $8 \times 125 = 1$k samples, denoted as $\mathcal{U}$. For comparison, we also randomly select 1k samples from the dataset, which shows similar distribution to the original distribution, denoted as $\mathcal{R}$.

Prompting Strategies: we use zero-shot prompting to assess LLMs' ability to identify individual linguistic structures in input text in a question-answering format, see Figure 2. We also investigate other prompting techniques, such as manually optimizing instructions, chain-of-thought (CoT) prompting (Wei et al., 2022) and structured prompting (Blevins et al., 2023). However, in a small scale experiment, the alternative approaches did not result in consistent performance improvement over the zero-shot approach. This could be because LLM's current pretraining does not fully capture the complex syntactic and semantic information of inputs required for fine-grained linguistic annotation. Instead, they might rely heavily on surface-level patterns, which limits the impact of more advanced prompting strategies.

4 Experimental Setup

Dataset & Evaluation: We use the CoNLL 2000 (Tjong Kim Sang and Buchholz, 2000) subset of the Penn Treebank corpus (Marcus et al., 1993) (Wall Street Journal (WSJ) sections 15, 16, 17, 18, 20), which provides ground truth POS tags and syn-

LLM	Sampling	P	R	F1
Llama3-70b	$\mathcal{R}$	**31.3**	**30.8**	**29.2**
Llama3-70b	$\mathcal{U}$	**28.2**	**27.5**	**26.1**
Llama3-8b	$\mathcal{R}$	24.0	26.8	23.2
Llama3-8b	$\mathcal{U}$	21.8	24.0	20.8
GPT-3.5	$\mathcal{R}$	21.6	26.1	21.2
GPT-3.5	$\mathcal{U}$	20.4	23.5	19.5
Llama2-70b	$\mathcal{R}$	13.4	21.4	14.7
Llama2-70b	$\mathcal{U}$	11.8	18.4	12.8
Mixtral-8x7b	$\mathcal{R}$	11.0	25.4	13.0
Mixtral-8x7b	$\mathcal{U}$	10.2	22.5	11.8
Mistral-7b	$\mathcal{R}$	7.4	15.2	8.0
Mistral-7b	$\mathcal{U}$	6.9	13.5	7.5
Llama2-7b	$\mathcal{R}$	7.0	9.8	7.4
Llama2-7b	$\mathcal{U}$	7.0	9.7	7.3
Gemini	$\mathcal{R}$	1.2	1.2	1.0
Gemini	$\mathcal{U}$	1.2	1.2	1.1

Table 1: Average performance in identifying linguistic structures. We compute precision, recall, and F1 for each sample, and average them across all samples to assess LLM performance in detecting linguistic structures.

tactic annotations. We use standard pre-processing to convert POS tags to Universal POS tags (Blevins et al., 2023). Following previous work (Blevins et al., 2023), we compute precision, recall, and F1 score for each sample, and average them across all samples to evaluate LLM performance in recognizing linguistic structures.

Large Language Models: We use several robust LLMs including GPT-3.5 (Ouyang et al., 2022) (`gpt-3.5-0613`), Gemini-Pro 1.0 (Team et al., 2023), Llama3 (7B, 13B, 70B) (Touvron et al., 2023), Llama2 (7B, 13B, 70B), and Mistral (7B, 8x7B) (Jiang et al., 2023, 2024).

5 Main Results

5.1 Deficient Linguistic Performance of LLMs

Tables 1 show significant performance differences between LLMs when tasked with identifying linguistic structures across different sampling strategies. Despite outperforming other LLMs by a large margin, Llama3-70b, Llama3-8b, and GPT-3.5 have considerably low performance in identifying linguistic structures. Among the evaluated LLMs, Llama3-70b performs the best, with average precision, recall, and F1 score of 31.3, 30.8, and 29.2 on randomly selected samples ($\mathcal{R}$), and 28.2, 27.5, and 26.1 on uniformly selected samples ($\mathcal{U}$). However, these results are substantially lower than that of traditional models with significantly smaller sizes (Manning et al., 2014).

In addition, Gemini, Llama-2 and Mistral show poor performance across all settings, indicating that many linguistic structures are indeed a blind spot for these LLMs. Larger scales of Llama2 and Mistral show slightly better performance, but still limited compared to GPT-3.5 and Llama3. These models often recognize the entire sentence as a phrase, can't distinguish between noun phrases (NPs) and verb phrases (VPs), and show poor performance in detecting clauses. Surprisingly, Gemini lacks the ability to identify linguistic structures, with an average F1 score close to 0. Through manual analysis, we find that Gemini often misinterprets linguistic queries with harmful content, see Section 6.2.

5.2 Task Complexity

We find all evaluated LLMs show stronger capability in detecting simpler linguistic structures (e.g. word-level) than more complex structures (e.g. sentence-level). Specifically, GPT-3.5 achieves an average F1 scores of 37.5 ($\mathcal{U}$) and 34.4 ($\mathcal{R}$) on word-level structures, but close to zero F1 on phrase-level and sentence-level structures, see Table 2. For some complex structures including verb phrase (VP), complex nominal (CN), dependent clause (DC), T-unit (T), and complex T-unit (CT), all LLMs have close to zero F1 score. This might be because these complex structures require a model to detect simpler structures (e.g. POS tags) and build on them in a compositional manner to correctly identify the more complex ones. Our results show that LLMs can accomplish simpler linguistic tasks but fail to perform complex ones, which mainly require knowledge about compositionality.

GPT-3.5 Performance: As shown in Table 2, word-level structures such as nouns, verbs, and punctuation are generally better annotated by GPT-3.5, while phrase-level and sentence-level structures, particularly verb phrases (VP), clauses (C), and complex T-units (CT), have significantly lower performance. These high-level structures are indeed blind spots for existing LLMs, due to their complexity and linguistic understanding required to accurately identify them. Overall, GPT-3.5 tends to perform better on $\mathcal{R}$ than on $\mathcal{U}$ across most word-level and phrase-level structures. Specifically, on randomly selected samples, GPT-3.5 achieves average F1 scores of 37.5, 4.0, and 0.0 on word-level, phrase-level, and sentence-level structures respectively. On $\mathcal{U}$, GPT-3.5 achieves lower average F1 scores of 34.4 for word-level structures, 4.2 for phrase-level structures, and remains at 0.0 for sentence-level structures. These results indicates the model's relative strength in handling word-level structures but its significant limitation on more complex structures.

Structure	Sampling	P	R	F1
Word-level Structure				
PUNC	$\mathcal{R}$	82.5	77.4	77.4
PUNC	$\mathcal{U}$	86.1	79.5	80.9
NOUN	$\mathcal{R}$	71.6	65.6	66.1
NOUN	$\mathcal{U}$	67.6	64.3	62.9
VERB	$\mathcal{R}$	61.4	61.4	55.9
VERB	$\mathcal{U}$	53.9	51.0	47.7
DET	$\mathcal{R}$	56.4	56.2	50.7
DET	$\mathcal{U}$	50.3	47.9	43.4
ADP	$\mathcal{R}$	48.7	60.1	50.2
ADP	$\mathcal{U}$	42.0	47.7	41.6
ADJ	$\mathcal{R}$	26.5	43.7	29.1
ADJ	$\mathcal{U}$	22.7	32.9	23.1
ADV	$\mathcal{R}$	25.1	37.0	26.6
ADV	$\mathcal{U}$	25.8	33.4	25.8
PRON	$\mathcal{R}$	18.0	35.1	20.1
PRON	$\mathcal{U}$	17.0	32.7	18.8
PRT	$\mathcal{R}$	8.5	34.7	12.7
PRT	$\mathcal{U}$	8.1	30.8	11.6
CONJ	$\mathcal{R}$	30.3	30.9	29.1
CONJ	$\mathcal{U}$	28.8	28.8	27.0
NUM	$\mathcal{R}$	31.5	29.8	29.7
NUM	$\mathcal{U}$	30.3	28.6	28.6
Average	$\mathcal{R}$	38.6	44.8	37.5
Average	$\mathcal{U}$	36.2	40.2	34.4
Phrase-level Structure				
ADVP	$\mathcal{R}$	6.0	22.8	8.1
ADVP	$\mathcal{U}$	6.5	20.4	7.9
NP	$\mathcal{R}$	11.5	14.0	11.8
NP	$\mathcal{U}$	12.3	14.1	12.2
ADJP	$\mathcal{R}$	1.2	5.9	1.8
ADJP	$\mathcal{U}$	1.7	5.8	2.1
VP	$\mathcal{R}$	2.2	3.3	2.3
VP	$\mathcal{U}$	2.7	3.6	2.7
CONJP	$\mathcal{R}$	0.0	0.0	0.0
CONJP	$\mathcal{U}$	0.0	0.0	0.0
CN	$\mathcal{R}$	0.0	0.1	0.0
CN	$\mathcal{U}$	0.0	0.0	0.0
Average	$\mathcal{R}$	3.5	7.7	4.0
Average	$\mathcal{U}$	3.9	7.4	4.2
Sentence-level Structure				
C	$\mathcal{R}$	0.1	0.3	0.1
C	$\mathcal{U}$	0.0	0.1	0.1
DC	$\mathcal{R}$	0.0	0.0	0.0
DC	$\mathcal{U}$	0.0	0.1	0.0
T	$\mathcal{R}$	0.0	0.0	0.0
T	$\mathcal{U}$	0.0	0.0	0.0
CT	$\mathcal{R}$	0.0	0.0	0.0
CT	$\mathcal{U}$	0.0	0.0	0.0
Average	$\mathcal{R}$	0.0	0.0	0.0
Average	$\mathcal{U}$	0.0	0.0	0.0

Table 2: Linguistic annotation performance of GPT-3.5 across different linguistic structure groups. We compute precision, recall, and F1 for each sample, and average them across all samples to assess LLM performance in detecting linguistic structures.

5.3 Linguistic Complexity

Performance Drop on Complexity-Balanced Samples: We observe significant differences in LLMs' performances on $\mathcal{R}$ and $\mathcal{U}$, as determined by a t-test at 95% confidence interval. All evaluated LLMs (GPT-3.5, Gemini, Llama3, Llama2, Mistral) show significant decrease in performance on uniformly selected samples ($\mathcal{U}$) compared to randomly selected ones ($\mathcal{R}$). The only exceptions are Gemini and Llama2-7B, which is likely due to their already low performance on both $\mathcal{R}$ and $\mathcal{U}$. For GPT-3.5, the performance drops from an F1 score of 21.2 to 19.5, with significant p-value of $1e$-7. We note that although the performance consistently and significantly decreases across models from R to U, the absolute drop is small to modest. This may be due to the already low overall performance ceiling on these tasks, where even small differences are meaningful; the models' relative robustness to certain types of linguistic complexity, despite persistent weaknesses on edge cases and harder structures; or the prevalence of easier (word-level) structures compared to more complex (phrase- or sentence-level) ones in the set of linguistic structures we investigate.

Linguistic Complexity Fluctuation: We find that LLMs' performance fluctuate with increasing linguistic complexity of inputs, as shown in Figure 3 for GPT-3.5; see performance of other LLMs in Appendix B Figures 9–15. Specifically, the performance of GPT-3.5 improves initially but then declines on structures like verbs, nouns, pronouns, adjectives, and adverbs as linguistic complexity increases, with F1 scores ranging from 0 to 50. This suggest that expert-defined linguistic complexity (Lu, 2010, 2012) may not align with how LLMs view complexity, which is an underexplored topic. Interestingly, for other structures like punctuation (PUNC), we observe the opposite performance trend. This is likely due to the unique nature of these linguistic structures as punctuation marks typically follow more predictable and less complex rules compared to other linguistic structures like verbs or nouns. In addition, performance trend vary substantially across different LLMs and scales. For instance, Llama3-70b consistently shows an inverted U-shaped ($\bigcap$) performance pattern, while

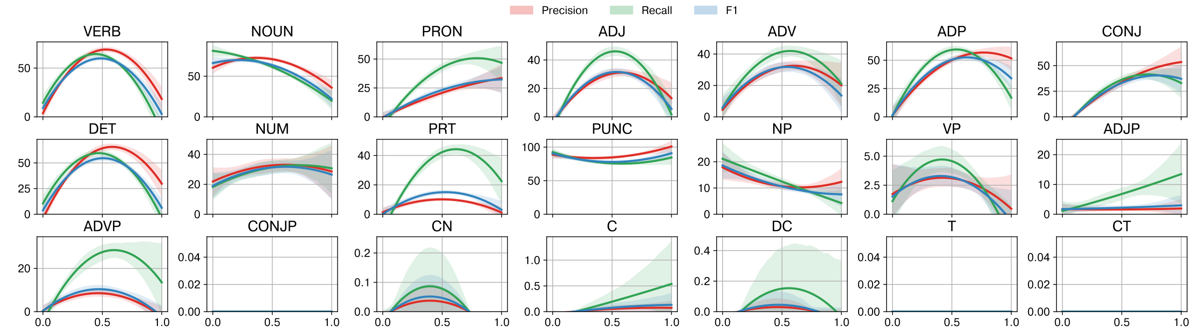


Figure 3: Performance of GPT-3.5 on texts of increasing linguistic complexity. GPT-3.5 achieves close to zero performance on CONJP, T, and CT. Figures 9-15 in Appendix B show results of other LLMs.

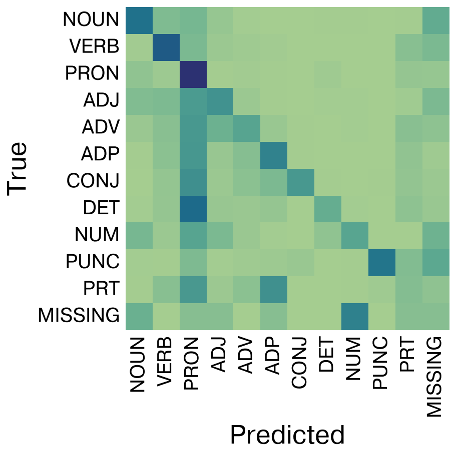


Figure 4: Confusion matrix of POS tagging on GPT-3.5. Darker indicates larger value. Diagonal/off-diagonal elements represent correct/wrong predictions respectively.

Llama2-70b have unique trends on noun and punctuation, which indicate model-specific challenges with different linguistic structures.

POS Tag Errors in GPT-3.5: Figure 4 shows a confusion matrix that assess the POS tags generated by GPT-3.5 Most of the errors stem from the model's failure to detect specific tags, denoted as "MISSING." The higher occurrence of MISSING cases is likely due to the increased complexity and linguistic knowledge required for these tasks–the need to identify and label all instances of linguistic structures in inputs. In addition, GPT-3.5 often confuses different POS tags with pronouns. This could be because pronouns often appear in diverse contexts where their function can be easily confused with other POS tags, such as determiners or nouns. In addition, GPT-3.5 (and other LLMs) tend to rely on surface-level patterns rather than deep linguistic understanding. Pronouns frequently co-occur within sentences, and the model may overgeneralize their patterns to other words.

Entity	NOUN	VERB	ADJ	ADV	ADP	CONJ	DET
# Dup.	334	370	93	156	526	400	635
# Succ.	0	1	0	0	5	1	2

Table 3: GPT-3.5 performance on samples that contain multiple instances of the same linguistic structure. Dup. indicates number of such texts (out of 1K) for each structure and Succ. indicates cases where *all* instances of the same POS tag are retrieved.

5.4 Multiple Structures and False Positives

When a samples contains multiple occurrences of the same linguistic structure, such as nouns, LLMs often struggle to retrieve all instances of of those structures. Table 3 shows that GPT-3.5 consistently fails to identify *all* nouns in any of the 334 samples containing more than one noun. This limitations extends beyond open-class words to closed-class tags such as prepositions (ADP), conjunctions (CONJ) and even determiners (DET).

We also observe that when a particular linguistic structure is absent in a given sample, LLMs still frequently make inaccurate predictions of its presence. Specifically for GPT-3.5, we find that in 6,892 out of 21,000 queries (33.9%), GPT-3.5 generates false positive predictions. Figure 5 shows the distribution of such errors across POS tag categories. The results show that GPT-3.5 often predict the existence of numerals (NUM), conjunctions (CONJ) and pronouns (PRON) when they are not present in the inputs. We conjecture that this behavior is due to biases in training data where certain words or structures co-occur frequently and the model learns to predict the presence of these words or structures based on relevant patterns in the training data, even when they don't exist in the input. For instance, if a sentence discusses quantities, the model might predict numerals. Therefore, false positive predictions

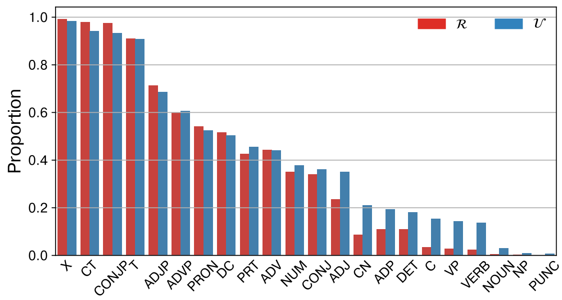


Figure 5: Distribution of false positive predictions by GPT-3.5 for absent linguistic structures in input. All evaluated LLMs show very similar distribution

for linguistic structures is common. In addition, all LLMs achieve higher recall than precision, especially all scales of LLaMA (see Table 1), again indicating that LLMs tend to retrieve more false positives than false negatives.

5.5 Model Capacity

We observe that models with higher capacity show slightly better performance. We evaluate the effect of model capacity, measured by the number of parameters, in performing fine-grained linguistic annotation tasks by comparing two scales of Llama3, Llama2, and Mistral, see Table 1. All models show improved or maintained linguistic performance as their capacity increases. However, it's noteworthy that the performance advantage may not be significant enough compared to the increase of scale. Specifically, using a 10 times larger Llama3 and Llama2 only boosts F1 score by 5.8 and 5.0, and 7.3 and 5.5 on randomly and uniformly sampled data respectively. The performance gain is also smaller on uniformly sampled inputs across all LLMs, due to the diverse inputs with various linguistic complexity, which outweighs model scale.

5.6 Dense model vs. Sparse model

Scaling up LLMs with Mixture-of-Experts (MoE) (Shazeer et al., 2017) in a sparse manner is a more efficient approach than dense scaling. We find that MoE can effectively boost LLM performance, see Mixtral 8x7b vs. Mistral-7b in Table 1. The performance of the MoE-based model–Mixtral 8x7b–is also comparable to that of Llama2-70b, a dense model of similar scale. This suggests that sparsity in LLMs is not a key or limiting factor in their fine-grained linguistic annotation ability.

6 Discussion

6.1 Limitations and Failure Cases

We identify significant limitations in the performance of the evaluated LLMs when responding to linguistic queries. Since GPT-3.5 is one of the most capable models with instruction-following capabilities in the evaluated LLMs, we mainly focus on its limitations. GPT-3.5 may **fail on linguistically easy examples**. For instance, it does not detect any of the 34 nouns that appear in the easiest linguistic examples in our dataset. GPT-3.5 (and most evaluated LLMs) may occasionally **skip tokens** in their responses. For example, they may skip tagging nouns or punctuation in inputs, which reduces their overall performance. They also **generate ill-formatted outputs**, including missing tags or corrupted parse trees. This is unexpected given that GPT-3.5 have a good knowledge about the definition of the linguistic tasks and required format; see Appendix A, Figures 6–8. GPT-3.5 (and most evaluated LLMs) may **generate biased outputs**. We find that the evaluated LLMs are biased to output common tokens and concepts, such as nouns and pronouns, while neglecting uncommon ones. Table 4 provides several example outputs. The first two show the tendency of GPT-3.5 to misclassify familiar structures such as nouns as adjectives or adverbs (first row) or verbs (second row). In addition, in may skip generating tags, see highlighted words "The" and "from" in Table 4. Note that the third example also shows GPT-3.5 mislabels many tokens, frequently replacing the correct POS tags with "PRON" or other incorrect tags.

6.2 Quality of Alignments

The LLMs we consider for this study have instruction-following capabilities. However, their performances in following linguistic-related instructions vary considerably. We find that GPT-3.5 tends to follow instructions better than other LLMs evaluated in our experiments. On the other hand, LLaMA-2 and LLaMA-3 generate irrelevant outputs including auxiliary text and special characters such as "\" and ">", even if we explicitly prompt them not to generate such characters. We note that the base model of GPT-3.5 has been fine-tuned on code data, which helps the model understand structured and instructional input. In addition, reinforcement learning from human feedback (RLHF) (Stiennon et al., 2020) is a more effective method to align LLMs for following instructions than fine-

Type	Expected output	Output by GPT-3.5
Easy example	He-PRON remains-VERB chief-ADJ executive-NOUN officer-NOUN .-PUNC	He-PRON remains-VERB chief-ADJ executive-ADJ officer-ADV .-PUNC
Confusion	The-DET consensus-NOUN calls-VERB for-ADP a-DET 0.5-NUM %-NOUN increase-NOUN in-ADP September-NOUN personal-ADJ income-NOUN and-CONJ a-DET 0.3-NUM %-NOUN gain-NOUN in-ADP consumption-NOUN .-PUNC	The-PRON consensus-NOUN calls-VERB for-PRON a-PRON 0.5-ADJ %-PUNC increase-VERB in-PRON September-NOUN personal-MISSING income-MISSING and-PRON a-PRON 0.3-ADJ %-PUNC gain-VERB in-PRON consumption-NOUN .-PUNC
Skip token	The-DET department-NOUN has-VERB collected-VERB over-ADV $-PUNC 6.5-NUM million-NUM from-ADP brokers-NOUN so-ADV far-ADV and-CONJ recommended-VERB more-ADJ than-ADP 30-NUM of-ADP them-PRON for-ADP criminal-ADJ prosecution-NOUN .-PUNC	The department-NOUN has-PRON collected-VERB over-PRON $-PRON 6.5-PRON million-PRON from brokers-NOUN so-PRON far-PRON and-PRON recommended-VERB more-PRON than-PRON 30-NOUN of-PRON them-PRON for-PRON criminal-PRON prosecution-PRON .-PUNC

Table 4: Summary of inaccurate, low quality and erroneous linguistic content generated by GPT-3.5.

tuning with conversation data (Zheng et al., 2023) and instruction tuning (Chung et al., 2022). In addition, GPT-3.5 has a much larger number of parameters, much larger capacity, than our other evaluated LLMs.

We find that Gemini rejects 24,630 prompts due to safety concerns related to sexual content, hate speech, harassment. Since we prompt Gemini to perform linguistic tasks, we conclude Gemini misinterpret linguistic queries with harmful contents, and it may have been superficially aligned (Zhou et al., 2023) to restrict its ability when processing sensitive words in the prompt. It is also over-aligned in terms of security-related content, leading to degenerated and undesired behavior.

Furthermore, we find that small scales of LLaMA-2 and Mistral do not follow instructions. They sometimes simply echo back the input sentence without linguistic annotation, responding they don't understand what the task is, or ask for the input to be processed. We hypothesize that this is strongly correlated with the distribution of instruction-tuning data, where linguistic instructions do not appear frequently.

6.3 Differences among Prompting Strategies

Prompting format and strategies differ in how they elicit knowledge from LLMs. However, we find that on identifying linguistic structures, adding in-context examples (Brown et al., 2020), CoT (Huang et al., 2022) or ReAct (Yao et al., 2023) provide only trivial performance gain over the plain prompt (0.05, 0.02 and 0.03 in F1 score respectively). We hypothesize that identifying linguistic structures, especially the complex ones, requires fundamental understanding of syntax and semantics, while CoT and ReAct focus on eliciting reasoning capabilities of LLMs, which is not sufficient.

6.4 Potential Solutions

Addressing the above limitations and biases requires developing effective data curation and training strategies using a linguist-in-the-loop process. Linguistically equitable and diverse datasets with balanced presence of linguistic structures (that specifically avoid overrepresentation of linguistically easy samples) are essential for NLP and for analyzing and understanding LLMs from a linguistic perspective. In what follows, we provide several avenues for investigating the above limitations.

Direct Training: Fine-tuning LLMs with targeted challenging examples, like those carrying complex sentence structures, or augmenting data to increase exposure to challenging examples can improve LLM's performance on fine-grained linguistic annotation tasks (Nguyen et al., 2024). The resulting computational costs can be alleviated through Parameter-Efficient Fine-Tuning techniques (Hu et al., 2022; Su et al., 2023).

Better Instructions: Designing linguistic instructions with sufficient context information to improve contextual understanding can potentially guide the model in handling complex structures. However, it would be challenging to generalize instructions to all linguistic structures and LLMs.

Curriculum Learning: LLM's performance on challenging linguistic structures could be improved by gradually training through a linguistic curriculum (Elgaar and Amiri, 2023a). A curriculum is a planned sequence of learning materials (a training paradigm) and an effective one can make learning efficient and effective for humans (Nishimura, 2018; Tabibian et al., 2019) and computers (Bengio et al., 2009). Curriculum learning techniques can present progressively increase the complexity of the linguistic structure of training samples, e.g. starting with easier structures before more complex ones to potentially improve LLM's performance on fine-grained linguistic annotation tasks.

Retrieval Augmented Generation: Incorporating documents with relevant linguistic knowledge retrieved from trustworthy sources can complement LLMs' knowledge (Lewis et al., 2020). For example, definitions of complex syntactic structures such as clauses and T-units can be retrieved to support more accurate analysis and generation. However, care must be taken to mitigate potential biases introduced within retrieval models (Ziems et al., 2024; Cheng and Amiri, 2025).

Tool Learning: LLMs can be trained to use tools (Schick et al., 2023), either by updating their parametric knowledge or interacting with tools directly. Training LLMs to use external linguistic tools, such as those discussed in this work (Lu, 2010, 2012; Lee et al., 2021; Lee and Lee, 2023), can potentially improve LLMs' capabilities on fine-grained linguistic tasks by complementing their internal representations with structured linguistic knowledge.

Human-in-the-Loop: Using a linguist-in-the-loop approach can provide a valuable feedback for refining model outputs. Expert input can help correct linguistic errors, mitigate biases, and guide the model toward more accurate and interpretable language understanding (Parrish et al., 2021).

7 Conclusion

We empirically study the ability of recent LLMs in annotating linguistic structures at different levels of linguistic complexity. Our study determines how accurately recent LLMs can detect complex linguistic structures in input text, which linguistic structures represent the blind spots of recent LLMs (the most challenging for LLMs), and how the performance of LLMs varies across different levels of linguistic complexity of inputs. Our findings show a tendency to overestimate the linguistic capabilities of LLMs in previous research, which mainly stems from the prevalence of linguistically easy examples in NLP datasets. To address this gap, we uniformly sample data from different linguistic complexity groups, to improve the reliability of evaluating LLMs' performance. Among all evaluated LLMs, Llama3-70b, Llama3-8b, and GPT-3.5 show relatively better performance in responding to linguistic queries–though overall performance remains low. We outline several potential solutions to address these limitations.

Limitations

Although we carefully developed and experimented with different prompting strategies, prompting cannot fully replace methods that directly analyze model's probability distributions over outputs (Hu and Levy, 2023; Kuribayashi et al., 2024). In addition, we did not investigate the ability of LLMs on a wider range of linguistic queries. For examples, linguistic structures related to *discourse complexity* (Feng et al., 2010; Guinaudeau and Strube, 2013; Bedi et al., 2015), which determines the complexity of higher-level structures and flow of language beyond individual phrases or sentences, need to investigated. Finally, understanding why a closed-source LLM produces a specific output can be challenging. This is a key challenge for deeper understanding of LLMs through theoretically-motivated linguistic probing techniques (Linzen et al., 2016; Warstadt et al., 2020; Hu et al., 2020), and limits our ability in providing insights into their potential weaknesses.

References

Ahmed Alajrami and Nikolaos Aletras. 2022. How does the pre-training objective affect what large language models learn about linguistic properties? In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 131–147, Dublin, Ireland. Association for Computational Linguistics.

Gillinder Bedi, Facundo Carrillo, Guillermo A Cecchi, Diego Fernández Slezak, Mariano Sigman, Natália B Mota, Sidarta Ribeiro, Daniel C Javitt, Mauro Copelli, and Cheryl M Corcoran. 2015. Automated analysis of free speech predicts psychosis onset in high-risk youths. *npj Schizophrenia*, 1(1):1–7.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 41–48, New York, NY, USA. Association for Computing Machinery.

Douglas Biber, Bethany Gray, Shelley Staples, and Jesse Egbert. 2020. Investigating grammatical complexity in l2 english writing research: Linguistic description versus predictive measurement. *Journal of English for Academic Purposes*, 46:100869.

Terra Blevins, Hila Gonen, and Luke Zettlemoyer. 2023. Prompting language models for linguistic structure. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6649–6663, Toronto, Canada. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris 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. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Angelica Chen, Ravid Shwartz-Ziv, Kyunghyun Cho, Matthew L Leavitt, and Naomi Saphra. 2024. Sudden drops in the loss: Syntax acquisition, phase transitions, and simplicity bias in MLMs. In *The Twelfth International Conference on Learning Representations*.

Jiali Cheng and Hadi Amiri. 2025. Equalizeir: Mitigating linguistic biases in retrieval models. In *Proceedings of the 2025 Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT‘s attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

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.

Nadir Durrani, Hassan Sajjad, Fahim Dalvi, and Yonatan Belinkov. 2020. Analyzing individual neurons in pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4865–4880, Online. Association for Computational Linguistics.

Mohamed Elgaar and Hadi Amiri. 2023a. HuCurl: Human-induced curriculum discovery. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1862–1877, Toronto, Canada. Association for Computational Linguistics.

Mohamed Elgaar and Hadi Amiri. 2023b. Ling-CL: Understanding NLP models through linguistic curricula. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13526–13542, Singapore. Association for Computational Linguistics.

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.

Lijun Feng, Noémie Elhadad, and Matt Huenerfauth. 2009. Cognitively motivated features for readability assessment. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 229–237, Athens, Greece. Association for Computational Linguistics.

Lijun Feng, Martin Jansche, Matt Huenerfauth, and Noémie Elhadad. 2010. A comparison of features for automatic readability assessment. In *Coling 2010: Posters*, pages 276–284, Beijing, China. Coling 2010 Organizing Committee.

Camille Guinaudeau and Michael Strube. 2013. Graph-based local coherence modeling. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 93–103, Sofia, Bulgaria. Association for Computational Linguistics.

Alex Housen, Bastien De Clercq, Folkert Kuiken, and Ineke Vedder. 2019. Multiple approaches to complexity in second language research. *Second language research*, 35(1):3–21.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Hai Hu, Kyle Richardson, Liang Xu, Lu Li, Sandra Kübler, and Lawrence Moss. 2020. OCNLI: Original Chinese Natural Language Inference. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3512–3526, Online. Association for Computational Linguistics.

Jennifer Hu and Roger Levy. 2023. Prompting is not a substitute for probability measurements in large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5040–5060, Singapore. Association for Computational Linguistics.

Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. 2022. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *Proceedings of the 39th International Conference on Machine Learning*.

Philip A. Huebner, Elior Sulem, Fisher Cynthia, and Dan Roth. 2021. BabyBERTa: Learning more grammar with small-scale child-directed language. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 624–646, Online. Association for Computational Linguistics.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Tatsuki Kuribayashi, Yohei Oseki, and Timothy Baldwin. 2024. Psychometric predictive power of large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1983–2005, Mexico City, Mexico. Association for Computational Linguistics.

Bruce W. Lee, Yoo Sung Jang, and Jason Lee. 2021. Pushing on text readability assessment: A transformer meets handcrafted linguistic features. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10669–10686, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Bruce W. Lee and Jason Lee. 2023. LFTK: Handcrafted features in computational linguistics. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 1–19, Toronto, Canada. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Ming Li, Yusheng Su, Hsiu-Yuan Huang, Jiali Cheng, Xin Hu, Xinmiao Zhang, Huadong Wang, Yujia Qin, Xiaozhi Wang, Kristen A. Lindquist, Zhiyuan Liu, and Dan Zhang. 2024. Language-specific representation of emotion-concept knowledge causally supports emotion inference. *iScience*, 27(12).

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Xiaofei Lu. 2010. Automatic analysis of syntactic complexity in second language writing. *International journal of corpus linguistics*, 15(4):474–496.

Xiaofei Lu. 2012. The relationship of lexical richness to the quality of ESL learners' oral narratives. *Mod. Lang. J.*, 96(2):190–208.

David Malvern, Brian Richards, Ngoni Chipere, and Pilar Durán. 2004. *Lexical diversity and language development*. Springer.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Dang Nguyen, Jiuhai Chen, and Tianyi Zhou. 2024. Multi-objective linguistic control of large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 4336–4347, Bangkok, Thailand. Association for Computational Linguistics.

Joel Nishimura. 2018. Critically slow learning in flashcard learning models. *Chaos*, 28(8):083115.

Lourdes Ortega. 2003. Syntactic complexity measures and their relationship to l2 proficiency: A research synthesis of college-level l2 writing. *Applied linguistics*, 24(4):492–518.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*.

Alicia Parrish, William Huang, Omar Agha, Soo-Hwan Lee, Nikita Nangia, Alexia Warstadt, Karmanya Aggarwal, Emily Allaway, Tal Linzen, and Samuel R. Bowman. 2021. Does putting a linguist in the loop improve NLU data collection? In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4886–4901, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Mandar Sharma, Nikhil Muralidhar, and Naren Ramakrishnan. 2023. Learning non-linguistic skills without sacrificing linguistic proficiency. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6178–6191, Toronto, Canada. Association for Computational Linguistics.

Noam Shazeer, *Azalia Mirhoseini, *Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*.

Sheng Shen, Zhewei Yao, Douwe Kiela, Kurt Keutzer, and Michael Mahoney. 2021. What's hidden in a one-layer randomly weighted transformer? In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2914–2921, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yikang Shen, Zhouhan Lin, Athul Paul Jacob, Alessandro Sordoni, Aaron Courville, and Yoshua Bengio. 2018. Straight to the tree: Constituency parsing with neural syntactic distance. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1171–1180, Melbourne, Australia. Association for Computational Linguistics.

Koustuv Sinha, Prasanna Parthasarathi, Joelle Pineau, and Adina Williams. 2021. UnNatural Language Inference. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7329–7346, Online. Association for Computational Linguistics.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems*.

Yusheng Su, Chi-Min Chan, Jiali Cheng, Yujia Qin, Yankai Lin, Shengding Hu, Zonghan Yang, Ning Ding, Xingzhi Sun, Guotong Xie, Zhiyuan Liu, and Maosong Sun. 2023. Exploring the impact of model scaling on parameter-efficient tuning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15062–15078, Singapore. Association for Computational Linguistics.

Shuo Sun, Yuchen Zhang, Jiahuan Yan, Yuze Gao, Donovan Ong, Bin Chen, and Jian Su. 2023. Battle of the large language models: Dolly vs LLaMA vs vicuna vs guanaco vs bard vs ChatGPT - a text-to-SQL parsing comparison. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11225–11238, Singapore. Association for Computational Linguistics.

Behzad Tabibian, Utkarsh Upadhyay, Abir De, Ali Zarezade, Bernhard Schölkopf, and Manuel Gomez-Rodriguez. 2019. Enhancing human learning via spaced repetition optimization. *Proc. Natl. Acad. Sci. U. S. A.*, 116(10):3988–3993.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Mildred C. Templin. 1957. *Certain Language Skills in Children: Their Development and Interrelationships*, ned - new edition edition, volume 26. University of Minnesota Press.

Erik F. Tjong Kim Sang and Sabine Buchholz. 2000. Introduction to the CoNLL-2000 shared task chunking. In *Fourth Conference on Computational Natural Language Learning and the Second Learning Language in Logic Workshop*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy. Association for Computational Linguistics.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the*

Association for Computational Linguistics, 8:377–392.

Jason Wei, Kelly Finn, Emma Templeton, Thalia Wheatley, and Soroush Vosoughi. 2021. Linguistic complexity loss in text-based therapy. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4450–4459, Online. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Orion Weller, Nicholas Lourie, Matt Gardner, and Matthew E. Peters. 2020. Learning from task descriptions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1361–1375, Online. Association for Computational Linguistics.

Kate Wolfe-Quintero, Shunji Inagaki, and Hae-Young Kim. 1998. Second language development in writing: Measures of fluency, accuracy, and complexity. *Honolulu, HI: University of Hawai'i, Second Language Teaching & Curriculum Center*.

Menglin Xia, Ekaterina Kochmar, and Ted Briscoe. 2016. Text readability assessment for second language learners. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 12–22, San Diego, CA. Association for Computational Linguistics.

Wei Xu, Chris Callison-Burch, and Courtney Napoles. 2015. Problems in current text simplification research: New data can help. *Transactions of the Association for Computational Linguistics*, 3:283–297.

Atsuki Yamaguchi, George Chrysostomou, Katerina Margatina, and Nikolaos Aletras. 2021. Frustratingly simple pretraining alternatives to masked language modeling. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3116–3125, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Songlin Yang and Kewei Tu. 2022. Bottom-up constituency parsing and nested named entity recognition with pointer networks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2403–2416, Dublin, Ireland. Association for Computational Linguistics.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A new dataset and method for automatically grading ESOL texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 180–189, Portland, Oregon, USA. Association for Computational Linguistics.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*.

Yaodong Yu, Sam Buchanan, Druv Pai, Tianzhe Chu, Ziyang Wu, Shengbang Tong, Benjamin David Haeffele, and Yi Ma. 2023. White-box transformers via sparse rate reduction. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Alla Zareva, Paula Schwanenflugel, and Yordanka Nikolova. 2005. Relationship between lexical competence and language proficiency: Variable sensitivity. *Studies in Second Language Acquisition*, 27(4):567–595.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2023. Lima: Less is more for alignment. *arXiv preprint arXiv:2305.11206*.

Caleb Ziems, William Held, Jane Dwivedi-Yu, and Diyi Yang. 2024. Measuring and addressing indexical bias in information retrieval. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12860–12877, Bangkok, Thailand. Association for Computational Linguistics.

A GPT's Knowledge on Target tasks

As illustrated in Figures 6–8, we asked relevant questions from GPT 3.5 about the target linguistic tasks of this study. The responses clearly indicate that GPT 3.5 have knowledge about the universal dependencies dataset and the universal POS tag set, the CoNLL 2000 shared task and its format, and the Penn Treebank dataset and the format of its syntactic structures.

B Detailed Results

B.1 Performance of all LLMs

We present the performances of all LLMs in Figure 9–15.

B.2 Performance across linguistic complexity spectrum

We present the performance-sample complexity plot for all models in Figure 9-Figure15. These results suggest that LLM's linguistic capability largely depends on the linguistic complexity of their inputs, and existing work may have overestimated the linguistic capability of LLMs. However, it's crucial to note that linguistic complexity as understood by humans and machines may differ significantly. Indeed, there are particular linguistic structures where model performance surprisingly escalates with increasing complexity. Understanding the reason for such model behaviors require understanding their learning dynamics and is the subject of our future work.

C Linguistic indices

Table 5 presents the 45 linguistic indices in our study.

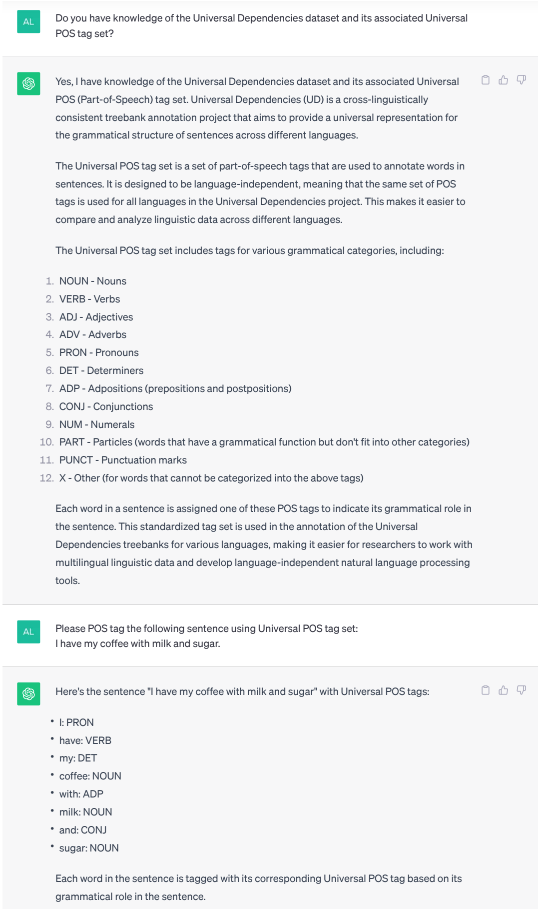
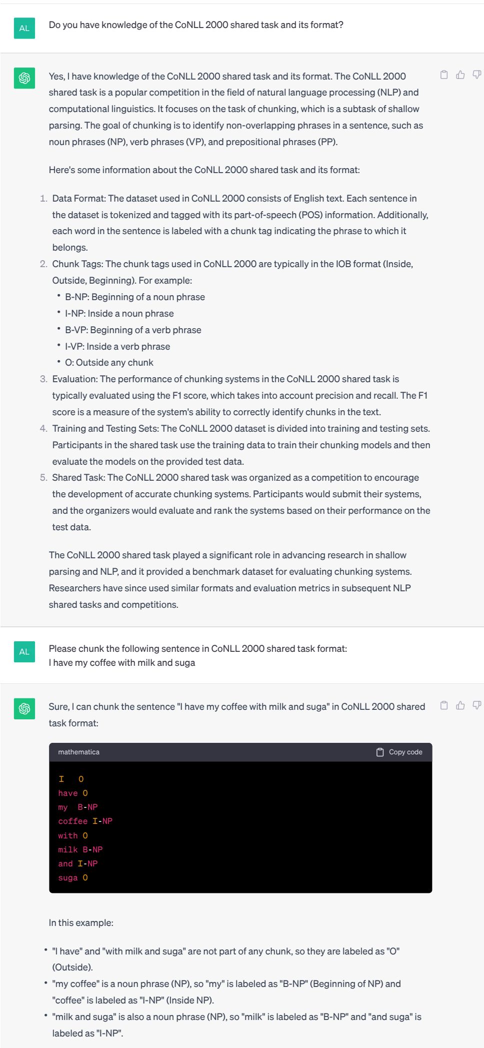


Figure 6: GPT's responses to our questions about universal dependencies dataset and its associated universal POS tag set.

Granularity	Name	Notation
Word	Nouns	Num_NN
Word	Verbs	Num_VB
Word	Adjectives	Num_JJ
Word	Adverbs	Num_RB
Word	Prepositions/Subordinates	Num_IN
Word	Coordinating Conjunction	Num_CC
Word	Determiner	Num_DT
Phrase	Noun Phrases	Num_NP
Phrase	Verb Phrases	Num_VP
Phrase	Adjective Phrases	Num_ADJP
Phrase	Adverb Phrases	Num_ADVP
Phrase	Preposition Phrases	Num_PP
Phrase	Conjunction Phrases	Num_CONJP
Phrase	Quantitative Phrases	Num_QP
Phrase	Complex Nominal	Num_CN
Sentence	T-Units	Num_T
Sentence	Complex T-Units	Num_CT
Sentence	Clause	Num_C
Sentence	Dependent Clause	Num_DC
Sentence	Fragment Clause	Num_FC

Table 5: Linguistic indices we use in the study.



AL Do you have knowledge of the CoNLL 2000 shared task and its format?

Yes, I have knowledge of the CoNLL 2000 shared task and its format. The CoNLL 2000 shared task is a popular competition in the field of natural language processing (NLP) and computational linguistics. It focuses on the task of chunking, which is a subtask of shallow parsing. The goal of chunking is to identify non-overlapping phrases in a sentence, such as noun phrases (NP), verb phrases (VP), and prepositional phrases (PP).

Here's some information about the CoNLL 2000 shared task and its format:

1. Data Format: The dataset used in CoNLL 2000 consists of English text. Each sentence in the dataset is tokenized and tagged with its part-of-speech (POS) information. Additionally, each word in the sentence is labeled with a chunk tag indicating the phrase to which it belongs.
2. Chunk Tags: The chunk tags used in CoNLL 2000 are typically in the IOB format (Inside, Outside, Beginning). For example:
   - B-NP: Beginning of a noun phrase
   - I-NP: Inside a noun phrase
   - B-VP: Beginning of a verb phrase
   - I-VP: Inside a verb phrase
   - O: Outside any chunk
3. Evaluation: The performance of chunking systems in the CoNLL 2000 shared task is typically evaluated using the F1 score, which takes into account precision and recall. The F1 score is a measure of the system's ability to correctly identify chunks in the text.
4. Training and Testing Sets: The CoNLL 2000 dataset is divided into training and testing sets. Participants in the shared task use the training data to train their chunking models and then evaluate the models on the provided test data.
5. Shared Task: The CoNLL 2000 shared task was organized as a competition to encourage the development of accurate chunking systems. Participants would submit their systems, and the organizers would evaluate and rank the systems based on their performance on the test data.

The CoNLL 2000 shared task played a significant role in advancing research in shallow parsing and NLP, and it provided a benchmark dataset for evaluating chunking systems. Researchers have since used similar formats and evaluation metrics in subsequent NLP shared tasks and competitions.

AL Please chunk the following sentence in CoNLL 2000 shared task format:
I have my coffee with milk and suga

Sure, I can chunk the sentence "I have my coffee with milk and suga" in CoNLL 2000 shared task format:

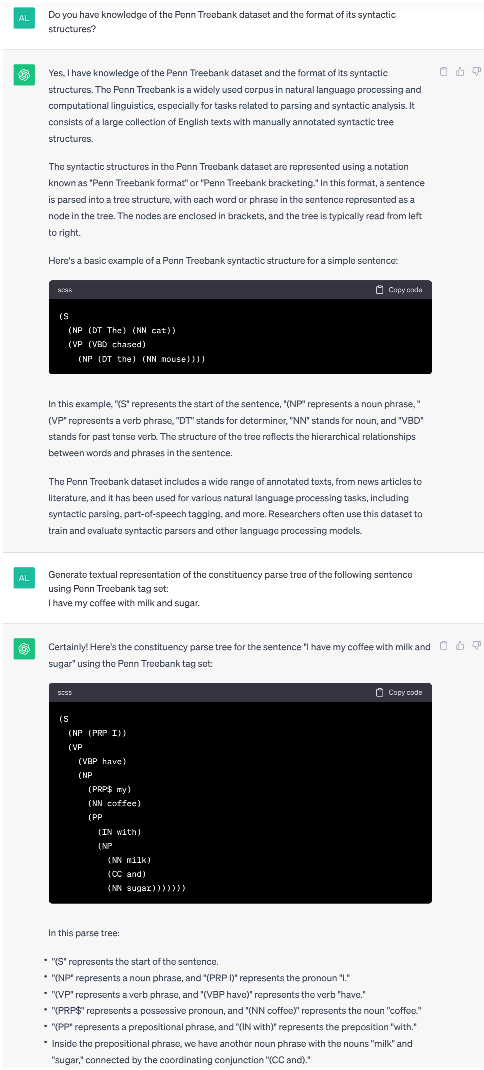
mathematica Copy code

```
I   O
have O
my  B-NP
coffee I-NP
with O
milk B-NP
and I-NP
suga O
```

In this example:

- "I have" and "with milk and suga" are not part of any chunk, so they are labeled as "O" (Outside).
- "my coffee" is a noun phrase (NP), so "my" is labeled as "B-NP" (Beginning of NP) and "coffee" is labeled as "I-NP" (Inside NP).
- "milk and suga" is also a noun phrase (NP), so "milk" is labeled as "B-NP" and "and suga" is labeled as "I-NP".

Figure 7: GPT's responses to our questions about the CoNLL 2000 shared task and its format.



AL Do you have knowledge of the Penn Treebank dataset and the format of its syntactic structures?

Yes, I have knowledge of the Penn Treebank dataset and the format of its syntactic structures. The Penn Treebank is a widely used corpus in natural language processing and computational linguistics, especially for tasks related to parsing and syntactic analysis. It consists of a large collection of English texts with manually annotated syntactic tree structures.

The syntactic structures in the Penn Treebank dataset are represented using a notation known as "Penn Treebank format" or "Penn Treebank bracketing." In this format, a sentence is parsed into a tree structure, with each word or phrase in the sentence represented as a node in the tree. The nodes are enclosed in brackets, and the tree is typically read from left to right.

Here's a basic example of a Penn Treebank syntactic structure for a simple sentence:

scss Copy code

```
(S
  (NP (DT The) (NN cat))
  (VP (VBD chased)
    (NP (DT the) (NN mouse))))
```

In this example, "(S" represents the start of the sentence, "(NP" represents a noun phrase, "(VP" represents a verb phrase, "DT" stands for determiner, "NN" stands for noun, and "VBD" stands for past tense verb. The structure of the tree reflects the hierarchical relationships between words and phrases in the sentence.

The Penn Treebank dataset includes a wide range of annotated texts, from news articles to literature, and it has been used for various natural language processing tasks, including syntactic parsing, part-of-speech tagging, and more. Researchers often use this dataset to train and evaluate syntactic parsers and other language processing models.

AL Generate textual representation of the constituency parse tree of the following sentence using Penn Treebank tag set:
I have my coffee with milk and sugar.

Certainly! Here's the constituency parse tree for the sentence "I have my coffee with milk and sugar" using the Penn Treebank tag set:

scss Copy code

```
(S
  (NP (PRP I))
  (VP
    (VBP have)
    (NP
      (PRP$ my)
      (NN coffee)
      (PP
        (IN with)
        (NP
          (NN milk)
          (CC and)
          (NN sugar)))))))
```

In this parse tree:

- "(S" represents the start of the sentence.
- "(NP" represents a noun phrase, and "(PRP I)" represents the pronoun "I."
- "(VP" represents a verb phrase, and "(VBP have)" represents the verb "have."
- "(PRP$" represents a possessive pronoun, and "(NN coffee)" represents the noun "coffee."
- "(PP" represents a prepositional phrase, and "(IN with)" represents the preposition "with."
- Inside the prepositional phrase, we have another noun phrase with the nouns "milk" and "sugar," connected by the coordinating conjunction "(CC and)."

Figure 8: GPT's responses to our questions about the Penn Treebank dataset and the format of its syntactic structures.

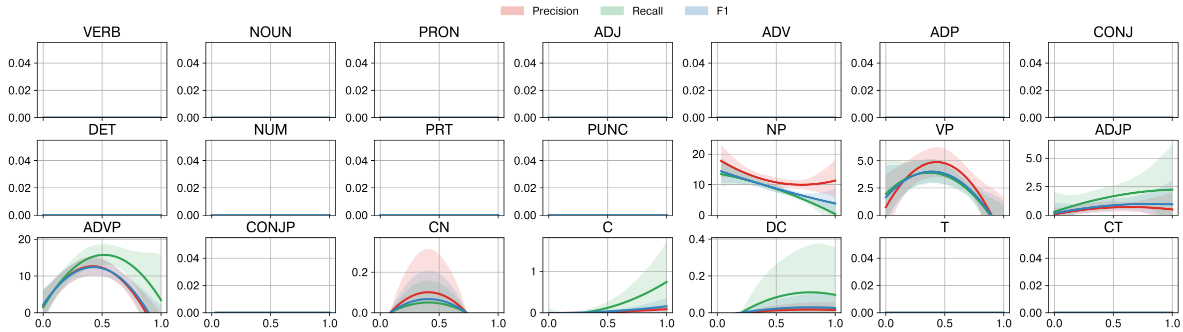


Figure 9: Performance of Gemini with respect to linguistic complexity.

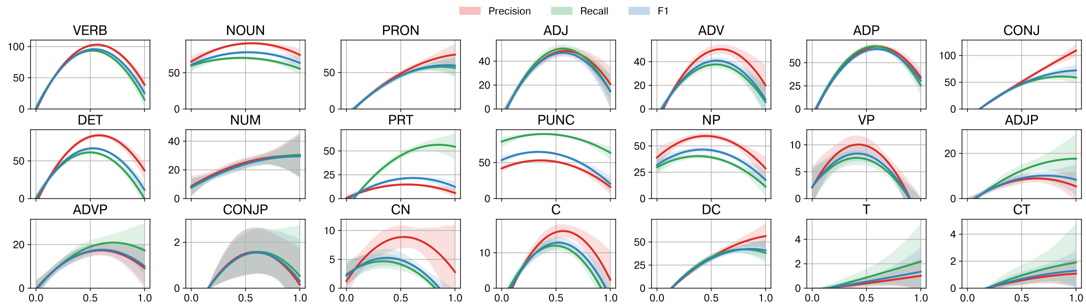


Figure 10: Performance of LLaMA3-70b with respect to linguistic complexity.

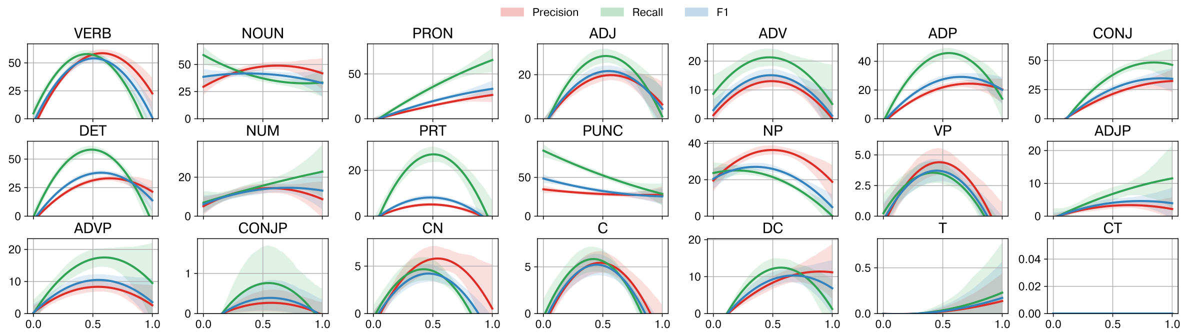


Figure 11: Performance of LLaMA2-70B with respect to linguistic complexity.

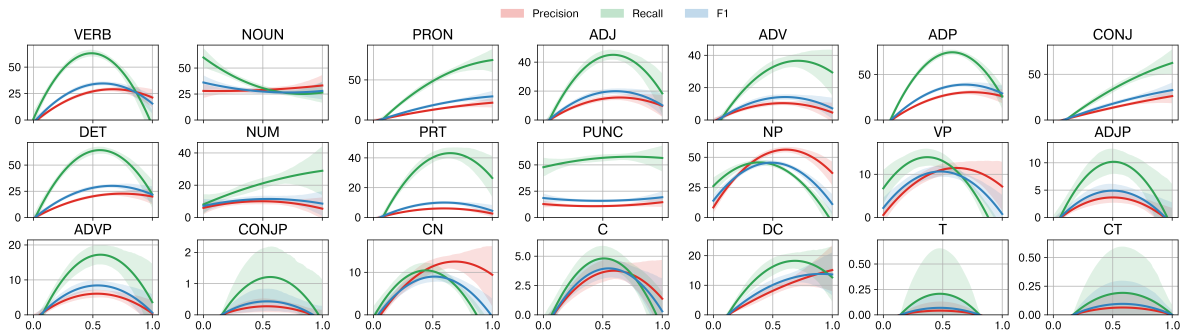


Figure 12: Performance of Mixtral-8x7B with respect to linguistic complexity.

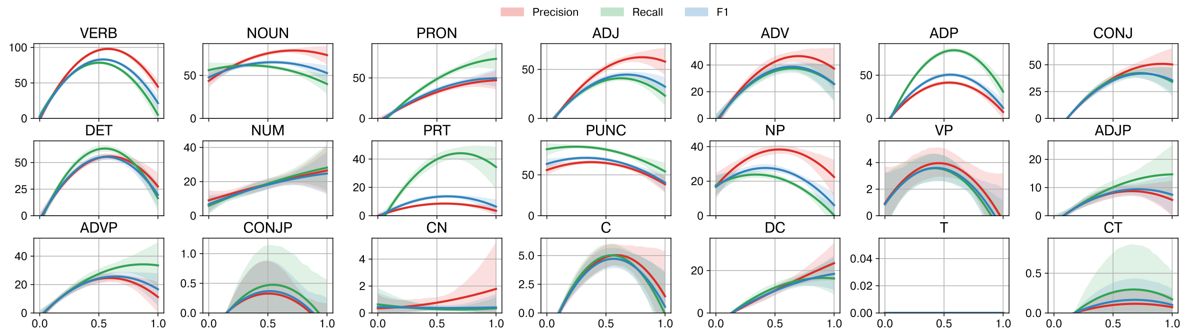


Figure 13: Performance of LLaMA3-8B with respect to linguistic complexity.

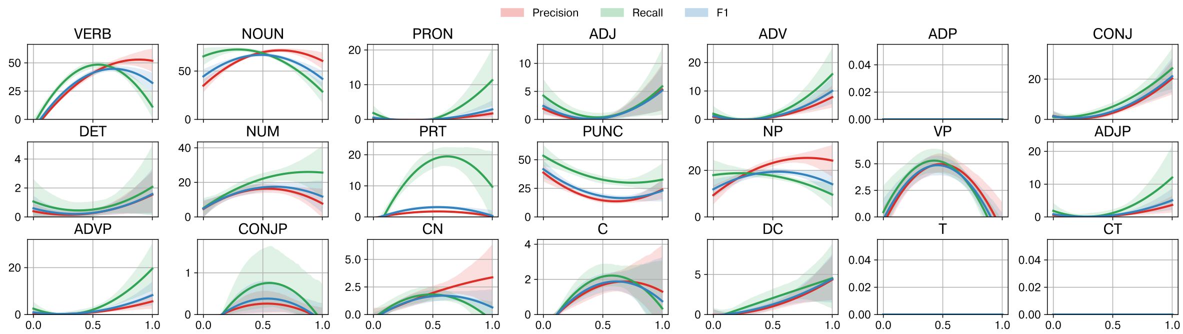


Figure 14: Performance of LLaMA2-7B with respect to linguistic complexity.

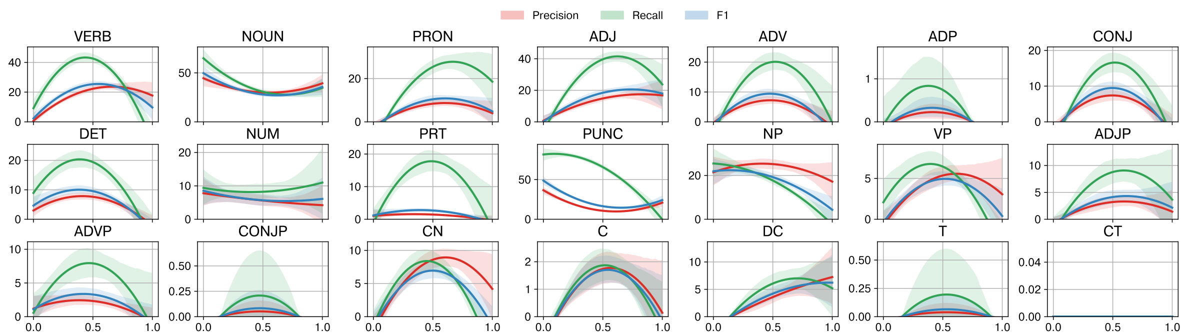


Figure 15: Performance of Mistral-7B with respect to linguistic complexity.