How to Make the Most of LLMs' Grammatical Knowledge for Acceptability Judgments

Yusuke Ide **Yuto Nishida** **Justin Vasselli** **Miyu Oba**
Yusuke Sakai **Hidetaka Kamigaito** **Taro Watanabe**
Nara Institute of Science and Technology
{ide.yusuke.ja6, nishida.yuto.nu8, vasselli.justin_ray.vk4, oba.miyu.ol2,
sakai.yusuke.sr9, kamigaito.h, taro}@is.naist.jp

Abstract

The grammatical knowledge of language models (LMs) is often measured using a benchmark of linguistic minimal pairs, where the LMs are presented with a pair of acceptable and unacceptable sentences and required to judge which is more acceptable. Conventional approaches directly compare sentence probabilities assigned by LMs, but recent large language models (LLMs) are trained to perform tasks via prompting, and thus, the raw probabilities they assign may not fully reflect their grammatical knowledge.

In this study, we attempt to derive more accurate acceptability judgments from LLMs using prompts and templates. Through extensive experiments in English and Chinese, we compare nine judgment methods and find two of them, a probability readout method—*in-template LP* and a prompt-based method—*Yes/No probability computing*, achieve higher accuracy than the conventional ones. Our analysis reveals that these methods excel in different linguistic phenomena, suggesting they access different aspects of LLMs' knowledge. We also find that ensembling the two methods outperforms single methods. Consequently, we recommend these techniques, either individually or ensembled, as more effective alternatives to conventional approaches for assessing grammatical knowledge in LLMs. [1]

1 Introduction

The grammatical knowledge of language models (LMs) is often measured using acceptability judgments (Lau et al., 2017; Warstadt et al., 2019). There are two main categories of acceptability judgment benchmarks, the single-sentence one and the minimal-pair (MP) one, as detailed in Section 2. We focus on the latter because it allows us to directly measure LMs' grammatical knowledge without task-specific fine-tuning. Below is an example of a minimal pair from Warstadt et al. (2020).

(a) *These casseroles disgust Kayla.*

(b) **These casseroles disgusts Kayla.*

Here, sentence (a) is acceptable or grammatically correct, while (b) is not, as its underlined verb violates the subject-verb agreement.

Meanwhile, the recent scaling up of model sizes and training data for LMs has made it possible to solve a wide range of tasks using few-shot or zero-shot prompting, without the need for task-specific fine-tuning (Brown et al., 2020; Liu et al., 2023), popularizing the term large language model (LLM). Incorporating learning techniques such as instruction tuning (Wei et al., 2022) and Direct Preference Optimization (DPO) (Rafailov et al., 2023) further improved the alignment of LLM outputs with human preferences. The LLMs trained by these techniques achieve good performance by prompting, i.e., guidance on what knowledge to elicit.

In this light, various methods can be developed to obtain more accurate acceptability judgments by providing LLMs with targeted guidance. However, as discussed in Section 2, most previous studies simply input the sentences into the (L)LM, calculate their probabilities, and consider the sentence with the higher probability as the acceptable one. Although Hu and Levy (2023) compared multiple methods of obtaining acceptability judgments from LLMs (see their Experiment 3b), their probability readout method and prompting method were limited to basic ones. As a result, they broadly claim that prompting is ineffective, whereas our experiments demonstrate that it can be highly effective with the proper technique.

We contribute to this area by comparing (1) the conventional sentence probability readout[2] methods, (2) our novel probability readout methods in

[1] Our codes and templates are published at https://github.com/Yusuke196/llm-acceptability.

[2] Readout refers to accessing an LLM's output layer to compute probabilities of strings (Kauf et al., 2024).

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 7416–7432
April 29 - May 4, 2025 ©2025 Association for Computational Linguistics

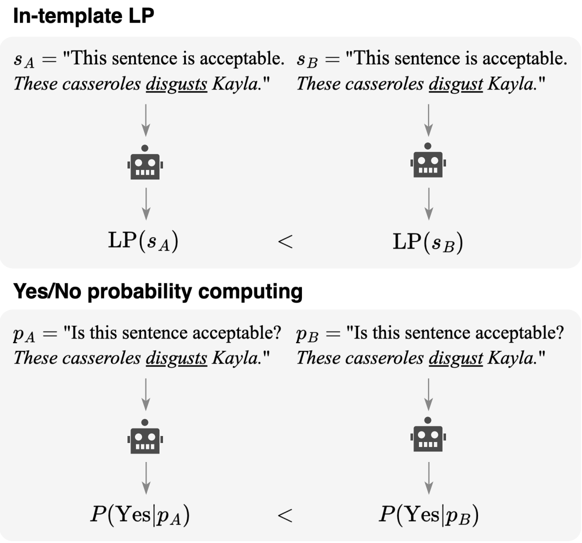

In-template LP
s_A = "This sentence is acceptable. These casseroles disgusts Kayla."
s_B = "This sentence is acceptable. These casseroles disgust Kayla."
LP(s_A) < LP(s_B)
Yes/No probability computing
p_A = "Is this sentence acceptable? These casseroles disgusts Kayla."
p_B = "Is this sentence acceptable? These casseroles disgust Kayla."
$P(\text{Yes}|p_A)$ < $P(\text{Yes}|p_B)$

Figure 1: Conceptual illustration of our top methods. Differences between paired sentences are underlined. Both methods judge the sentence that results in a higher (log) probability acceptable. See Section 3 for the details.

in-template settings, and (3) prompt-based methods. In the in-template probability readout, we insert each sentence into a template and use the LLM to calculate the probability of the complete string. The template allows us to guide the LLM to judge the sentence's grammaticality in a way that conventional probability readouts cannot. We call the most basic method *in-template LP*, where LP stands for log probability. For prompt-based methods, we investigate a basic method of asking LLMs to respond with a choice and *Yes/No probability computing (Yes/No prob comp)*, where we compute the normalized probability of "Yes" versus "No", inspired by UniEval (Zhong et al., 2022). Figure 1 presents the conceptual illustration of in-template LP and Yes/No prob comp.

To rigorously compare these methods, we conduct experiments using eight LLMs and two MP benchmarks (one for English and one for Chinese). The results show the effectiveness of the two methods. In-template LP consistently outperforms conventional methods, achieving the highest accuracies on the Chinese benchmark. Yes/No prob comp achieves the highest accuracies on the English benchmark in all but one setting.

Moreover, our analysis demonstrates the following key findings. (1) In-template LP and Yes/No prob comp have different strengths; for example, Yes/No prob comp is robust against token-length bias. This indicates that they access different aspects of LLMs' grammatical knowledge, contributing to a more comprehensive evaluation. (2) Ensembling the two methods further improves the accuracy, revealing their complementary capabilities. The highest score, achieved with Qwen2, is 1.6 percentage points higher than humans on the English benchmark. Based on these findings, we recommend the following: if possible, ensemble the two methods; otherwise, use in-template LP. (3) We identify a common weakness across all our combinations of the LLMs and methods: they struggle to make correct judgments on linguistic phenomena where the unacceptable sentence can be obtained by shuffling the words in the acceptable one, which presents a challenge for future work.

2 Related Work

Benchmarks of acceptability judgments can be divided into two categories: single-sentence benchmarks and MP benchmarks. The single-sentence benchmarks pose a binary classification of single sentences as seen in CoLA (Warstadt et al., 2019), a dataset composed of sentences each labeled acceptable or unacceptable. CoLA was incorporated into the natural language understanding benchmark GLUE (Wang et al., 2018) and has been used to evaluate a wide range of models, including LMs. However, single-sentence benchmarks cannot measure LMs' grammatical knowledge directly because they require training a supervised classifier before the evaluation. This makes it difficult to distinguish between the knowledge of the model itself and what is learned by training the classifier (Warstadt et al., 2020).

In contrast, MP benchmarks present minimally different pairs and the task is to determine which is the most acceptable sentence, eliminating the need for a classifier. As another advantage of the MP benchmark, minimal pairs can be automatically generated in a controlled manner, providing a sufficient amount of quality data for model evaluation (Linzen et al., 2016). In conventional experiments using an MP benchmark, judgments are made based on sentence probabilities. Models are evaluated by whether they assign a higher probability to the acceptable sentence in each minimal pair. This method, which we call sentence probability readout, has been dominantly employed for MP acceptability judgments across languages (Marvin and Linzen, 2018; Warstadt et al., 2020; Mueller et al., 2020; Haga et al., 2024; Xiang et al., 2021;

Someya and Oseki, 2023, inter alia).

Experiments with prompting LLMs have been conducted on both single-sentence benchmarks and MP benchmarks. Zhang et al. (2024) compared various models, including LLMs, on a single-sentence benchmark. On MP benchmarks, Hu and Levy (2023) compared the sentence probability readout and prompting. However, their probability readout and prompt-based methods relied on basic implementations, without systematically exploring alternative ways to calculate sentence probabilities or optimize prompting strategies.

Another line of work has studied biases that affect the performance of sentence probability readout. For example, it is known that the sentence probabilities given by LMs tend to decline as the token length of the sentence grows; normalized measures such as PenLP (Wu et al., 2016) have been shown to mitigate some of this bias (Lau et al., 2020), but they do not eliminate it (Ueda et al., 2024).

3 Methods

We compare three different groups of methods to extract acceptability judgments from the LLMs.

3.1 Sentence Probability Readout

In sentence probability readout, we input each sentence of a given pair into a model to obtain the probabilities assigned to each token. The probabilities are then used to compute a probability score for each sentence, and the sentence given the higher score is predicted to be acceptable.

We experiment with three measures to compute the probability scores: LP, MeanLP, and PenLP. All of them have been widely used in acceptability judgments.[3] LP is the unnormalized log probability of the sentence

$$\mathrm{LP}(\boldsymbol{s}) = \log P(\boldsymbol{s}) \tag{1}$$

where $\boldsymbol{s}$ is the input sequence of tokens and $P(\boldsymbol{s})$ is the probability assigned to $\boldsymbol{s}$ by the model

$$P(\boldsymbol{s}) = \prod_{i=1}^{|\boldsymbol{s}|} P(t_i|t_{<i}). \tag{2}$$

Because LP tends to get smaller as the sentence gets longer (Ueda et al., 2024), we also compute two normalized measures, MeanLP and PenLP (Lau et al., 2020; Wu et al., 2016),

$$\mathrm{MeanLP}(\boldsymbol{s}) = \frac{\log P(\boldsymbol{s})}{|\boldsymbol{s}|} \tag{3}$$

$$\mathrm{PenLP}(\boldsymbol{s}) = \frac{\log P(\boldsymbol{s})}{((5+|\boldsymbol{s}|)/(5+1))^{\alpha}} \tag{4}$$

where α is a hyperparameter to scale the sentence length, reducing the impact of long sentences and ensuring a fair comparison across different lengths. We set $\alpha = 0.8$ following Lau et al. (2020) and Ueda et al. (2024). We hereafter refer to the three judgment methods simply by the name of the corresponding measures: *LP*, *MeanLP*, *PenLP*.

3.2 In-template Probability Readout

In-template probability readout follows the same steps of computing and comparing probabilities as sentence probability readout. Meanwhile, its input string is built by embedding the sentences in a template designed to draw focus to their grammaticality. The input has two types: *in-template single* and *in-template comparative*. For each type, we prepare five templates per language because the performance can vary due to minor differences in expressions within prompts (Zheng et al., 2023). The templates were created based on those of Flan[4](Wei et al., 2022). For Chinese experiments, we use translations of English templates. Translations were generated by DeepL[5] and post-edited by a native Chinese speaker.

In-template single In-template single templates have one placeholder where the target sentence is inserted. Table 1 shows an example input.

As the length of the input changes depending on the length of the target sentence, predictions by in-template single inputs must also use normalization techniques. We thus apply each of the three measures explained above to the method, dubbing the corresponding methods *in-template LP*, *in-template MeanLP*, and *in-template PenLP*, respectively. The final measure depends on whether we base our computations on the whole input string or the target sentence only. We report the result of the former because it performed better in our preliminary experiments.

In-template comparative In-template comparative inputs are built by filling two placeholders; we

[3] SLOR (Kann et al., 2018) is also commonly used, but it requires building a unigram model using the training data of the LM. Because the training data of the LLMs we use is not publicly available, we skip the acceptability measure.

[4] https://github.com/google-research/FLAN

[5] https://www.deepl.com

Input Type	Example Input
Sentence	*Many girls insulted themselves.*
In-template single	The following sentence is grammatically acceptable.\n\n*Many girls insulted themselves.*
In-template comparative	The following sentence A is grammatically acceptable while B is not.\n\nA: *Many girls insulted themselves.*\nB:*Many girls insulted herself.*

Table 1: Example English inputs of the readout methods. The target or inserted sentences are in italics. See Table 6 for Chinese versions.

Type	Role	Example Message
A/B	System	Your task is to compare the quality of given sentences.
	User	One of the following sentences is grammatically acceptable and the other is not. Which one is acceptable? Respond with A or B as your answer.\n\nA: *Many girls insulted themselves.*\nB: *Many girls insulted herself.*
Yes/No	System	Your task is to evaluate the quality of given text.
	User	Is the following sentence grammatically acceptable? Respond with Yes or No as your answer.\n\n*Many girls insulted themselves.*

Table 2: Example English messages for prompting. The target or inserted sentences are in italics. See Table 7 for Chinese versions.

insert the target sentence into the first one and the other sentence of the minimal pair into the second. Table 1 shows an example input. Note that the second sentence is supplementary, and our aim here is to measure the acceptability of the first one.

In-template comparative does not need normalization, because the sum of the token lengths of two sentences and, thus, the length of the whole input string is constant, no matter which of the paired sentences comes first. Hence, we only calculate LP for the in-template comparative input, referring to this method as *in-template comparative LP*.

3.3 Prompt-based Methods

In prompt-based methods, *A/B prompting* and *Yes/No prob comp*, we provide the models with prompts that include a question. For both methods, we prepare a system message and a user message. The system message describes the task to be solved, which has been shown to enhance the performance (Peng et al., 2023). The user message includes the main prompt, and we prepare five versions of each method's prompt template per language. Each user message is built by inserting one or two sentences into a template, as we do for in-template probability readout. When prompting a base model, we concatenate the two messages and append the string `\nAnswer:` at the end. When prompting an instruct model, we apply chat templates[6] to maximize the performance, including control tokens like `<|begin_of_text|>` in the inputs to the model. For Chinese experiments, we use translations of English templates verified by a native Chinese speaker.

A/B prompting A/B prompting inputs a prompt containing the paired sentences to the models and asks which sentence is acceptable. The prompt is exemplified in Table 2. The user message contains one acceptable and one unacceptable sentence. Their order (which sentence goes to A or B) is randomized to eliminate the potential bias from the order (Pezeshkpour and Hruschka, 2023). We perform constrained decoding by outlines[7] (Willard and Louf, 2023) to ensure that the model outputs either A or B, because our preliminary experiments without outlines observed many outputs violating the constraint. We turn off sampling in decoding.

Yes/No probability computing In Yes/No prob comp, we compute the score of each sentence as the normalized probability of "Yes" versus "No" given a prompt asking its acceptability. An example prompt is shown in Table 2. We predict the sentence that resulted in a higher "Yes" probability to be acceptable. This method is inspired by UniEval (Zhong et al., 2022), which is shown to correlate well with human judgments in evaluat-

[6] https://huggingface.co/docs/transformers/en/chat_templating

[7] https://github.com/outlines-dev/outlines

ing natural language generation. We formulate the probability given a sentence s as follows,

$$P(\text{“Yes”}|s) = \frac{P_{\text{LLM}}(\text{“Yes”}|s)}{P_{\text{LLM}}(\text{“Yes”}|s) + P_{\text{LLM}}(\text{“No”}|s)} \tag{5}$$

where $P_{\text{LLM}}(\cdot)$ is the probability of a word assigned by the model. For the Chinese experiments, we substitute “是” and “否” for “Yes” and “No”, respectively, if these words are not segmented into subwords; otherwise, we employ the same formulation as English experiments.

4 Experimental Setup

Models We use eight state-of-the-art LLMs, among which Llama-3-70B (Meta, 2024), Mixtral-8x7B-v0.1 (Jiang et al., 2024), Qwen2-57B-A14B (Qwen Team, 2024), and Yi-1.5-34B (Young et al., 2024) are base models, while Llama-3-70B-Instruct, Mixtral-8x7B-Instruct-v0.1, Qwen2-57B-A14B-Instruct, and Yi-1.5-34B-Chat are instruct models based on the pre-trained counterparts. These models ranked relatively high in the leaderboard of English language understanding[8] or Chinese LLMs[9] at the time of model selection. For Chinese experiments, we substituted Yi-1.5 for Mixtral because Mixtral is not explicitly trained for Chinese tasks, while Yi-1.5 ranked high in the Chinese LLM leaderboard. We hereafter abbreviate these models, e.g., to Llama-3, omitting the model sizes and minor versions. Post-training for the three instruct models includes supervised fine-tuning on an instruction dataset, i.e., instruction-tuning and DPO. They are publicly available on Hugging Face Hub. On inference, we perform 4-bit quantization using bitsandbytes[10] to compress the models. The computational budgets are described in Appendix C.

Benchmarks We use two MP acceptability judgment benchmarks: BLiMP (Warstadt et al., 2020) for English and CLiMP (Xiang et al., 2021) for Chinese. BLiMP is composed of minimal pairs from 67 different paradigms, each containing 1,000 pairs of sentences. The paradigms are grouped into 12 categories of linguistic phenomena. CLiMP consists of 16 paradigms, each with 1,000 pairs like BLiMP. The paradigms are grouped into 9 linguistic phenomena. We focus on these two because no other MP benchmarks contain hundreds or thousands of sentences for each paradigm, to our knowledge, which is important for reliable experiments. The linguistic phenomena and licenses of the two benchmarks are detailed in Appendix B.1 and Appendix B.2, respectively.

Evaluation metric We evaluate the methods by accuracy. Random chance accuracy is 50%, as the two classes are balanced in our benchmarks.

5 Results

Table 3 summarizes the results. The statistics of the in-template probability readout methods and prompting methods are the average of the five scores by the five versions of templates.[11]

Comparison between methods reveals the effectiveness of in-template LP and Yes/No prob comp. In-template LP achieves significantly higher accuracies than LP in all settings, i.e., benchmark-model pairs, across languages. On CLiMP, it marks the highest accuracy for all models. Sentence probability readout methods—LP, MeanLP, and PenLP—underperformed in-template LP in all settings, although they have been dominant in previous studies. This indicates that including guidance about the task in the input to LLMs improves judgment performance. Meanwhile, Yes/No prob comp achieves the highest accuracy for five out of six models on BLiMP; the mean accuracies of Llama-3-Instruct and Qwen2 exceed that of humans (the majority vote of 20 crowd workers) reported in Warstadt et al. (2020), 88.6%.

Methods giving two sentences to the model—A/B prompting and in-template comparative LP—consistently underperformed Yes/No prob comp and in-template LP, respectively, suggesting that LLMs struggle to handle choice identifiers such as A and B (See Appendix D.2 for more analysis, which reveals the low performance of A/B prompting can be partly attributed to a preference for a specific choice). This may suggest that making LLMs select an identifier from multiple choices—a common approach for classification tasks such as question answering (Hendrycks et al., 2021)—is

[8] https://paperswithcode.com/sota/multi-task-language-understanding-on-mmlu

[9] https://huggingface.co/spaces/BAAI/open_cn_llm_leaderboard

[10] https://github.com/TimDettmers/bitsandbytes

[11] We conducted paired bootstrap resampling (Koehn, 2004) to validate whether each of our methods statistically significantly improves accuracy compared to LP, the best-performing conventional method; we performed 1,000 resamplings with replacement, sampling 67,000 and 16,000 instances each time for BliMP and CLiMP, respectively.

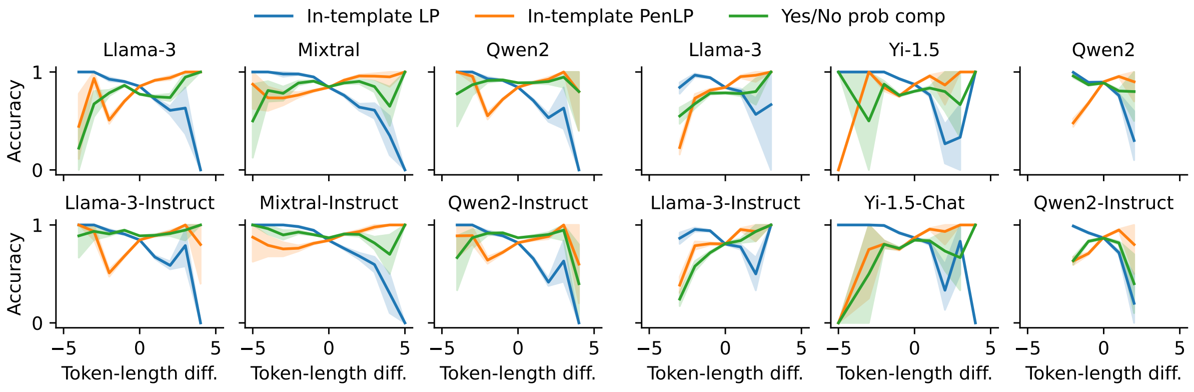
	Llama-3	Llama-3-Inst.	Mixtral	Mixtral-Inst.	Qwen2	Qwen2-Inst.
LP	79.6	77.1	82.5	82.3	80.4	79.7
MeanLP	77.1	74.8	79.6	79.4	77.7	77.1
PenLP	79.2	76.8	82.2	82.0	79.9	79.2
In-template LP	$\mathbf{84.4}^{*}_{\pm 0.5}$	$\underline{83.5}^{*}_{\pm 0.5}$	$\underline{84.0}^{*}_{\pm 0.5}$	$\underline{83.5}^{*}_{\pm 0.9}$	$\underline{83.9}^{*}_{\pm 0.3}$	$80.1^{*}_{\pm 1.0}$
In-template MeanLP	$82.6^{*}_{\pm 0.7}$	$81.9^{*}_{\pm 0.5}$	$82.6^{*}_{\pm 0.3}$	$82.2_{\pm 0.8}$	$82.0^{*}_{\pm 0.7}$	$78.7_{\pm 1.1}$
In-template PenLP	$\underline{83.8}^{*}_{\pm 0.5}$	$83.0^{*}_{\pm 0.5}$	$83.8^{*}_{\pm 0.4}$	$83.3^{*}_{\pm 1.0}$	$83.2^{*}_{\pm 0.4}$	$79.8_{\pm 1.1}$
In-template compar. LP	$71.8_{\pm 4.5}$	$61.8_{\pm 2.6}$	$72.1_{\pm 3.2}$	$68.4_{\pm 1.2}$	$62.7_{\pm 3.7}$	$58.5_{\pm 3.8}$
A/B prompting	$77.4_{\pm 3.6}$	$81.9^{*}_{\pm 3.7}$	$76.5_{\pm 4.3}$	$80.5_{\pm 3.5}$	$80.8^{*}_{\pm 1.1}$	$\underline{82.5}^{*}_{\pm 0.3}$
Yes/No prob comp	$73.6_{\pm 3.2}$	$\mathbf{88.9}^{*}_{\pm 0.3}$	$\mathbf{84.1}^{*}_{\pm 1.2}$	$\mathbf{84.0}^{*}_{\pm 2.0}$	$\mathbf{89.0}^{*}_{\pm 0.2}$	$\mathbf{86.8}^{*}_{\pm 0.4}$

(a) BLiMP

	Llama-3	Llama-3-Inst.	Yi-1.5	Yi-1.5-Chat	Qwen2	Qwen2-Inst.
LP	$\underline{83.2}$	80.4	$\underline{86.8}$	$\underline{85.3}$	85.4	$\underline{85.4}$
MeanLP	74.5	71.7	75.9	74.5	74.5	74.3
PenLP	80.3	77.7	84.3	82.3	82.2	82.0
In-template LP	$\mathbf{85.7}^{*}_{\pm 0.5}$	$\mathbf{82.9}^{*}_{\pm 0.3}$	$\mathbf{87.4}^{*}_{\pm 0.4}$	$\mathbf{86.8}^{*}_{\pm 0.8}$	$\mathbf{87.9}^{*}_{\pm 0.3}$	$\mathbf{86.2}^{*}_{\pm 0.3}$
In-template MeanLP	$79.9_{\pm 0.9}$	$77.7_{\pm 0.6}$	$78.8_{\pm 1.3}$	$79.4_{\pm 1.2}$	$77.7_{\pm 1.2}$	$77.5_{\pm 1.4}$
In-template PenLP	$83.0_{\pm 0.5}$	$\underline{80.7}^{*}_{\pm 0.4}$	$84.0_{\pm 0.8}$	$83.3_{\pm 1.1}$	$83.4_{\pm 0.4}$	$82.9_{\pm 0.5}$
In-template compar. LP	$68.2_{\pm 2.4}$	$58.2_{\pm 3.0}$	$63.9_{\pm 4.6}$	$61.8_{\pm 3.4}$	$68.1_{\pm 4.5}$	$60.6_{\pm 3.9}$
A/B prompting	$68.0_{\pm 3.5}$	$69.2_{\pm 4.0}$	$74.2_{\pm 3.6}$	$75.5_{\pm 3.0}$	$77.3_{\pm 4.2}$	$80.9_{\pm 1.6}$
Yes/No prob comp	$76.3_{\pm 1.3}$	$76.9_{\pm 0.9}$	$78.2_{\pm 0.9}$	$81.7_{\pm 0.3}$	$\underline{87.2}^{*}_{\pm 0.4}$	$83.9_{\pm 0.4}$

(b) CLiMP

Table 3: Percentage accuracy (averaged over templates) by method and model. ± denotes standard deviation. The bold font denotes the best scores. Underlines denote the second best. See Appendix D.1 for the max accuracy. * denotes scores significantly higher than LP ($p \leq 0.01$).



(a) BLiMP

(b) CLiMP

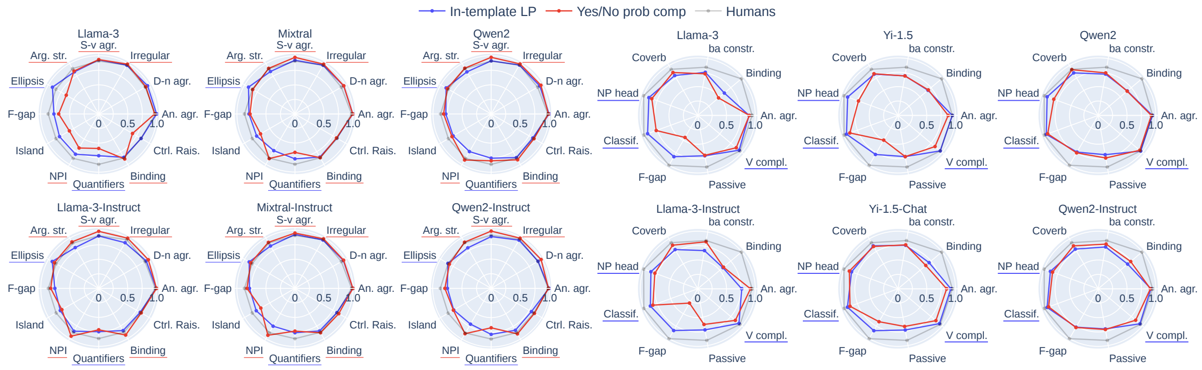
Figure 2: Top methods' correlation between the token-length difference ($|s_{\text{acceptable}}| - |s_{\text{unacceptable}}|$) and the accuracy (best template) by model, showing the robustness of Yes/No prob comp. The shadow denotes 95% confidence intervals.

suboptimal for making full use of LLMs' knowledge in general; substituting our choice-free methods could improve performance in other tasks.

6 Analysis

Given the remarkable performance of Yes/No prob comp and in-template LP, this section further investigates their strengths and weaknesses.

Yes/No prob comp is robust against token-length bias. Figure 2 illustrates the correlations between the token-length difference and the accuracy. The token-length difference is $|s_{\text{acceptable}}| - |s_{\text{unacceptable}}|$ where s_{x} denotes the token sequence of either sentence. A level line denotes that the token-length difference does not affect the method. Across the models, the following trends are observed. (1) The token-length difference biases the readout methods. The accuracy of in-template LP



(a) BLiMP

(b) CLiMP

Figure 3: Accuracy of top methods (best template) and humans, by linguistic phenomenon and model, demonstrating the difference in strengths between methods. For each benchmark, phenomena where either method wins the other by at least 1 point for at least five models are underlined.

	BLiMP	CLiMP
In-template LP	−0.118	−0.135
In-template PenLP	0.094	0.182
Yes/No prob comp	**−0.019**	**0.051**

Table 4: Top methods' point biserial correlation coefficient between the prediction success and token-length difference (averaged over models) by benchmark. The bold font denotes the value closest to zero.

decreases as the difference grows because the acceptable sentence is less likely to be given a high probability. In-template PenLP suffers a reversed tendency; due to normalization, it becomes weaker as the unacceptable sentence gets longer than the acceptable one. (2) Yes/No prob comp is relatively robust against the bias. Its accuracy does not drop as much as that of the other methods, even when the token lengths differ by a large margin.

These observations are quantitatively supported by the correlation coefficient between the token-length difference and the dichotomous variable that gets 1 for a successful prediction and 0 for a failure. Table 4 shows the average coefficients of Yes/No prob comp are much closer to zero than those of readout methods on both benchmarks, demonstrating its robustness against the token-length bias. This, in turn, indicates that the readout methods need better normalization techniques.

In-template readout and Yes/No prob comp excel in different phenomena. Figure 3 illustrates the accuracy of in-template LP, Yes/No prob comp, and the humans by linguistic phenomenon; the scores of humans are from Warstadt et al. (2020) and Xiang et al. (2021). For in-template LP and Yes/No prob comp, the result of the best-performing template is shown. Here we find that the two methods have different strengths. On BLiMP, Yes/No prob comp excels at phenomena such as Subject-verb agreement (S-v agr.) and Binding for most (at least five out of six) models (See Appendix B.1 for examples of these phenomena). In contrast, in-template LP is superior in Ellipsis and Quantifiers for most models. On CLiMP, Yes/No prob comp is good at Coverb and in-template LP at NP head finality (NP head). This indicates that each method harnesses different aspects of the models' grammatical knowledge.

Given the aforementioned token-length bias, one hypothesis to explain this difference would be that Yes/No prob comp is more accurate in phenomena with large token-length differences. Our analysis on BLiMP, however, does not support this, as detailed in Appendix D.3.

Meanwhile, some phenomena are challenging for both methods. As Figure 3 shows, on BLiMP, the two methods underperform humans for all models in Island effects and Quantifiers, which were shown to be challenging also by Warstadt et al. (2020). On CLiMP, our methods struggle with phenomena such as Binding and Passive.

Voting ensembles of the top two methods further improve the performance. Given the different strengths of in-template LP and Yes/No prob comp, we ensemble these methods to see if they can complement each other to achieve higher accuracy.

Now we have 10 sets of predictions by the two methods, as each has five templates. To compare ensembling single-method predictions and ensembling multi-method predictions on equal terms, we

	BLiMP				CLiMP			
	Llama-3	Llama-3 -Instruct	Qwen2	Qwen2 -Instruct	Llama-3	Llama-3 -Instruct	Qwen2	Qwen2 -Instruct
Ensemble L0:P5	76.0	89.0	89.5	87.0	76.0	76.6	87.6	84.1
Ensemble L2:P3	82.7	**89.7**	**90.2**	**87.7**	79.1	78.4	89.4	87.1
Ensemble L3:P2	**86.1**	86.3	86.7	84.8	**87.1**	**83.8**	**90.5**	**89.4**
Ensemble L5:P0	85.1	84.1	84.3	81.0	86.1	83.0	88.4	87.0
In-template LP (oracle)	85.0	84.2	84.1	81.3	86.1	83.2	88.2	86.5
Yes/No prob comp (oracle)	77.8	89.3	89.2	87.4	78.1	78.4	87.5	84.3

Table 5: Percentage accuracy of voting ensembles of in-template LP and Yes/No prob comp, with the oracle (max) accuracy by single methods (best template). The bold font denotes the best ensemble score. Results on Mixtral and Yi models are omitted because the same trend is observed; see Appendix D.4 for their results.

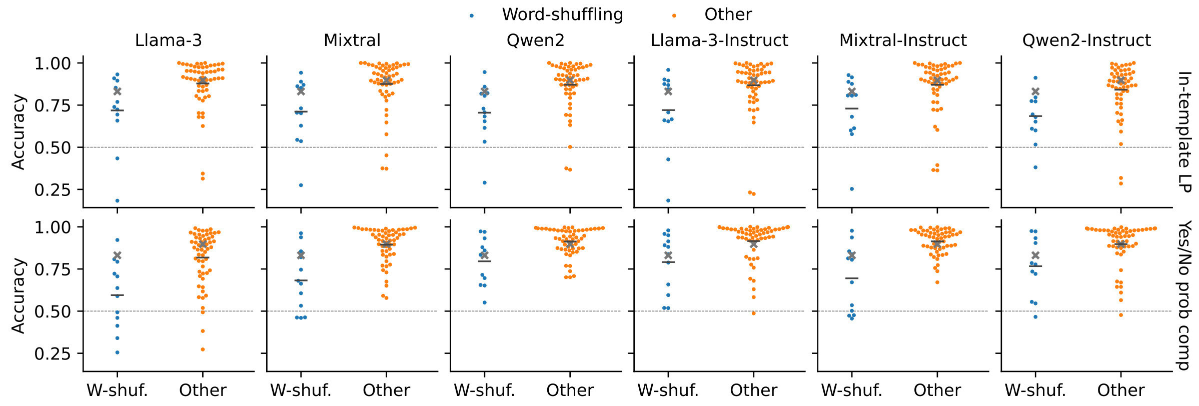


Figure 4: Accuracy on BLiMP by paradigm type, method, and model (best template), showing the difficulty of word-shuffling paradigms. Each dot represents a paradigm, short bars denote the mean accuracy of the category, cross markers denote mean human accuracies, and dashed lines denote the chance accuracy.

sample five without replacement from the 10 and perform majority voting by the five. We prepare the following four settings, which differ in the balance between the two methods: *P-only*, *Mix-P3*, *Mix-L3*, and *L-only*. P-only and L-only are ensembles of predictions by Yes/No prob comp only and in-template LP only, respectively. Mix-P3 and Mix-L3 use three predictions from Yes/No prob comp and in-template LP, respectively, with two predictions from the other method. We report the mean accuracy of 10 trials for these settings as the result is non-deterministic due to sampling.

Table 5 demonstrates that ensembles of the two methods, either Mix-P3 or Mix-L3, yield the best results across models, surpassing the oracle (max) accuracies of methods without ensembling, except for Mixtral-Instruct on BLiMP. The highest score by Mix-P3 with Qwen2 is 1.6 points higher than humans (described in Section 5). This indicates that the two methods have complementary capabilities.

Word-shuffling paradigms are challenging. BLiMP's 67 paradigms can be divided into two categories based on whether paired sentences of the paradigm have the same bag of words when the cases are ignored. We call the paradigms where this is true *word-shuffling paradigms*. In other words, the unacceptable sentence in word-shuffling paradigms can be obtained by shuffling the words in the acceptable counterpart. Here, we focus on BLiMP because CLiMP does not have word-shuffling paradigms. Following is an example pair from a word-shuffling paradigm, coordinate_structure_constraint_complex_left_branch.

(a) *What reports did Rose hate and James find?*

(b) **What did Rose hate reports and James find?*

Figure 4 shows the accuracy by paradigm, paradigm type—word-shuffling or not, demonstrating that the word-shuffling paradigms have much lower accuracy than other phenomena across methods and models. The accuracy of word-shuffling paradigms averaged over models and methods is 71.6% compared to 87.9% of other paradigms. The best accuracy on word-shuffling paradigms, achieved by Yes/No prob comp with Qwen2, was

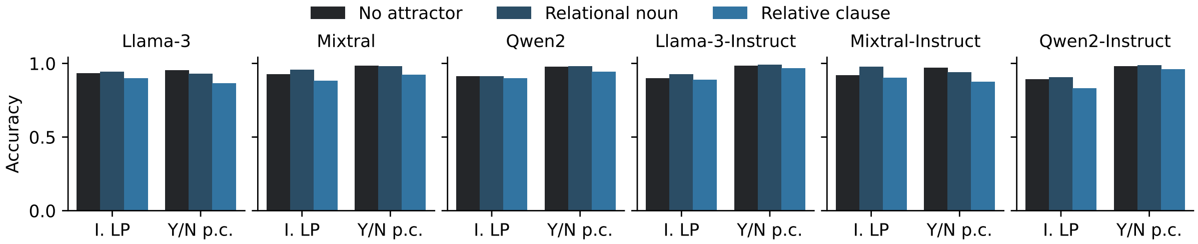


Figure 5: Accuracy on S-v agr. paradigms in BLiMP by attractor type, method, and model (best template), showing that attractors in a relative clause hinder acceptability judgments.

79.6%, which is much lower than that of the same method on other paradigms, 91.3%. Note that such large differences are not observed for humans according to the data released by Warstadt et al. (2020); humans' accuracy on word-shuffling paradigms and other paradigms are, on average, 83.1% and 89.7%, respectively. This suggests that word-shuffling paradigms remain a challenge for the current LLMs, as they have trouble recognizing word shuffling that corrupts grammar even with our best-performing method.

Why are word-shuffling paradigms difficult? We hypothesize that the LLMs are insensitive to the word order of the inputs. Previously, Sinha et al. (2021b) demonstrated the word-order insensitivity of Transfer-based models, such as RoBERTa (Liu et al., 2019) and BART (Lewis et al., 2020), in the task of natural language inference. Sinha et al. (2021a) and Pham et al. (2021) report the results to the same effect. The Transformer architecture, which forms the basis of the models they and we used, may generally struggle to capture the difference in word order.

Attractors in a relative clause lower the performance. Attractors refer to material intervening agreement dependencies, and previous work has shown that they can impair acceptability judgments. Below are examples of different attractor types in S-v agr., from Warstadt et al. (2020); (a) contains no attractor, (b) has an attractor as a relational noun, and (c) has an attractor in a relative clause. Because subject-verb agreement does not exist in Chinese, we focus on English here.

(a) *The sisters bake/*bakes.*
(b) *The sisters of Cheryl bake/*bakes.*
(c) *The sisters who met Cheryl bake/*bakes.*

Using such sentence pairs, Warstadt et al. (2020) and Mueller et al. (2020) investigated the sensitivity of models to mismatches in S-v agr. They showed an attractor noun of the opposite number often deteriorates accuracy, particularly when the attractor is a relational clause, as in sentence (c).

Figure 5 shows both top methods suffer the same issue across models. The accuracy averaged over methods and models drops from 94.5% for the agreement with no attractors to 90.4% for the agreement with attractors in a relative clause. In contrast, attractors as relational nouns do not necessarily lower the performance.

7 Conclusion

We investigate how to derive the most accurate acceptability judgments from LLMs by comparing nine methods, using eight LLMs and two benchmarks. Our experiments reveal that in-template LP consistently outperforms conventional sentence probability readout methods while Yes/No prob comp achieves the highest accuracies on the English benchmark. Our analysis demonstrates that the two methods excel in different phenomena, suggesting they harness different aspects of LLMs' grammatical knowledge. We also find that ensembling the two methods achieves even higher accuracy. Consequently, we recommend ensembling the two methods or employing in-template LP as more effective alternatives to conventional approaches. Meanwhile, we show that word-shuffling paradigms remain difficult for all our methods, posing a challenge for future work.

8 Limitations

In BLiMP and CLiMP, acceptable sentences are designed to be grammatical or well-formed, and the sentences in each paradigm were validated by humans (Warstadt et al., 2020; Xiang et al., 2021). However, acceptable sentences can be non-sensical as they are automatically generated without their senses being considered. For example, the acceptable sentence in the following pair from the existential_there_quantifiers_2 phenomenon is nonsensical; it is difficult to identify what situation is described in the acceptable sentence, at least without context.

(a) *All convertibles weren't there existing.*
(b) **There weren't all convertibles existing.*

To make correct acceptability judgments, our prompts or in-template inputs guided the LLMs to focus on acceptability or grammatical correctness. However, this may be insufficient when making judgments on phenomena that contain many nonsensical acceptable sentences like the above. To address this issue, future work could use prompts or in-template inputs that clearly instruct the model to focus only on the form of each sentence and ignore whether the sentence is sensical.

One of the key findings of this paper is that in-template LP and Yes/No prob comp excel in different linguistic phenomena. To investigate the reasons for the differences, we examined a hypothesis that Yes/No prob comp is more accurate in phenomena where the acceptable sentence is, on average, longer than the unacceptable one (See Appendix D.3). Yet the hypotheses were not supported, leaving the cause of their different strengths an open question.

Throughout the paper, we focused on experiments in the zero-shot setting, aligning the conditions with conventional probability readout methods. It is notable that some methods nonetheless achieved accuracies surpassing humans. However, providing few-shot examples in in-template LP and Yes/No prob comp might increase accuracy even further, which is worth investigating in future work.

Additionally, this paper focused on English and Chinese acceptability judgments. Although this was inevitable to secure a sufficient data size per benchmark, experiments in more languages are desired for more robust conclusions.

Acknowledgments

We are grateful to Hongyu Sun for verifying the Chinese translations of the prompts and templates.

References

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.

Akari Haga, Saku Sugawara, Akiyo Fukatsu, Miyu Oba, Hiroki Ouchi, Taro Watanabe, and Yohei Oseki. 2024. Modeling overregularization in children with small language models. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 14532–14550, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

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.

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, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts. *Preprint*, arXiv:2401.04088.

Katharina Kann, Sascha Rothe, and Katja Filippova. 2018. Sentence-level fluency evaluation: References help, but can be spared! In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 313–323, Brussels, Belgium. Association for Computational Linguistics.

Carina Kauf, Emmanuele Chersoni, Alessandro Lenci, Evelina Fedorenko, and Anna A. Ivanova. 2024. Comparing plausibility estimates in base and instruction-tuned large language models. *Preprint*, arXiv:2403.14859.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Jey Han Lau, Carlos Armendariz, Shalom Lappin, Matthew Purver, and Chang Shu. 2020. How furiously can colorless green ideas sleep? sentence acceptability in context. *Transactions of the Association for Computational Linguistics*, 8:296–310.

Jey Han Lau, Alexander Clark, and Shalom Lappin. 2017. Grammaticality, acceptability, and probability: A probabilistic view of linguistic knowledge. *Cogn. Sci.*, 41(5):1202–1241.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

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.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9).

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. *Preprint*, arXiv:1907.11692.

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium. Association for Computational Linguistics.

Meta. 2024. Introducing meta llama 3: The most capable openly available llm to date.

Aaron Mueller, Garrett Nicolai, Panayiota Petrou-Zeniou, Natalia Talmina, and Tal Linzen. 2020. Cross-linguistic syntactic evaluation of word prediction models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5523–5539, Online. Association for Computational Linguistics.

Keqin Peng, Liang Ding, Qihuang Zhong, Li Shen, Xuebo Liu, Min Zhang, Yuanxin Ouyang, and Dacheng Tao. 2023. Towards making the most of ChatGPT for machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5622–5633, Singapore. Association for Computational Linguistics.

Pouya Pezeshkpour and Estevam Hruschka. 2023. Large language models sensitivity to the order of options in multiple-choice questions. *Preprint*, arXiv:2308.11483.

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, Online. Association for Computational Linguistics.

Qwen Team. 2024. Introducing qwen1.5.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*.

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 *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2888–2913, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Koustuv Sinha, Prasanna Parthasarathi, Joelle Pineau, and Adina Williams. 2021b. 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.

Taiga Someya and Yohei Oseki. 2023. JBLiMP: Japanese benchmark of linguistic minimal pairs. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1581–1594, Dubrovnik, Croatia. Association for Computational Linguistics.

Naoya Ueda, Masato Mita, Teruaki Oka, and Mamoru Komachi. 2024. Token-length bias in minimal-pair paradigm datasets. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16224–16236, Torino, Italia. ELRA and ICCL.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: A benchmark of linguistic minimal pairs for English. In *Proceedings of the Society for Computation in Linguistics 2020*, pages 409–410, New York, New York. Association for Computational Linguistics.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Brandon T. Willard and Rémi Louf. 2023. Efficient guided generation for large language models. *Preprint*, arXiv:2307.09702.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *Preprint*, arXiv:1609.08144.

Beilei Xiang, Changbing Yang, Yu Li, Alex Warstadt, and Katharina Kann. 2021. CLiMP: A benchmark for Chinese language model evaluation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2784–2790, Online. Association for Computational Linguistics.

Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Tao Yu, Wen Xie, Wenhao Huang, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Pengcheng Nie, Yuchi Xu, Yudong Liu, Yue Wang, Yuxuan Cai, Zhenyu Gu, Zhiyuan Liu, and Zonghong Dai. 2024. Yi: Open foundation models by 01.ai. *Preprint*, arXiv:2403.04652.

Ziyin Zhang, Yikang Liu, Weifang Huang, Junyu Mao, Rui Wang, and Hai Hu. 2024. MELA: Multilingual evaluation of linguistic acceptability. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2658–2674, Bangkok, Thailand. Association for Computational Linguistics.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. 2024. Large language models are not robust multiple choice selectors. In *The Twelfth International Conference on Learning Representations*.

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. *Preprint*, arXiv:2306.05685.

Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. 2022. Towards a unified multi-dimensional evaluator for text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2023–2038, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

A Templates and Prompts

Input Type	Example Input
Sentence	王玉珍震惊了她自己。
In-template single	以下句子在语法上是可以接受的。\n\n王玉珍震惊了她自己。
In-template comparative	下面的句子A在语法上是可以接受的，而B则不可以。\n\n A：王玉珍震惊了她自己。\nB：王玉珍震惊了他自己。

Table 6: Example Chinese inputs of the readout methods. The target or inserted sentences are underlined. See Table 1 for English versions.

Type	Role	Example Message
A/B	System	您的任务是评估给定文本的质量。
	User	下列句子中，一个符合语法要求，另一个不符合语法要求。哪个可以接受？请用A或B作答。\n\nA：王玉珍震惊了她自己。\nB：王玉珍震惊了他自己。
Yes/No	System	您的任务是评估给定文本的质量。
	User	下面的句子符合语法要求吗？请用“是”或“否”回答。\n\n王玉珍震惊了她自己。

Table 7: Example Chinese messages for prompting. The target or inserted sentences are underlined. See Table 2 for English versions.

B Benchmarks

B.1 Linguistic Phenomena

Field	Phenomenon	Acceptable Example	Unacceptable Example
Morphology	Anaphor agr.	*Many girls insulted themselves.*	*Many girls insulted herself.*
	Det.-noun agr.	*Rachelle had bought that chair.*	*Rachelle had bought that chairs.*
	Irregular forms	*Aaron broke the unicycle.*	*Aaron broken the unicycle.*
	Subject-verb agr.	*These casseroles disgust Kayla.*	*These casseroles disgusts Kayla.*
Syntax	Arg. structure	*Rose wasn't disturbing Mark.*	*Rose wasn't boasting Mark.*
	Ellipsis	*Anne's doctor cleans one important book and Stacey cleans a few.*	*Anne's doctor cleans one book and Stacey cleans a few important.*
	Filler-gap	*Brett knew what many waiters find.*	*Brett knew that many waiters find.*
	Island effects	*Which bikes is John fixing?*	*Which is John fixing bikes?*
Semantics	NPI licensing	*The truck has clearly tipped over.*	*The truck has ever tipped over.*
	Quantifiers	*No boy knew fewer than six guys.*	*No boy knew at most six guys.*
Syn. & Sem.	Binding	*Carlos said that Lori helped him.*	*Carlos said that Lori helped himself.*
	Control/raising	*There was bound to be a fish escaping.*	*There was unable to be a fish escaping.*

Table 8: Minimal pairs from each of the twelve linguistic phenomena covered by BLiMP. Differences are underlined.

Phenomenon	Acceptable Example	Unacceptable Example
Anaphor agreement	王玉珍 震惊-了 她自己。 Jane.F shock-PST herself. *’Jane shocked herself.’*	王玉珍 震惊-了 他自己。 Jane.F shock-PST himself. *’Jane shocked himself.’*
Binding	杨颖 治疗 吴宇涛 之后 佩服-过 她自己。 Yang.F cure Wu.M after admire-PST herself *’Yang admired herself after she cured Wu.’*	杨颖 治疗 吴宇涛 之后 佩服-过 他自己。 Yang.F cure Wu.M after admire-PST himself *’Yang admired himself after she cured Wu.’*
bǎ construction	王鑫 把 自行车 扔 了。 Wong.M BA bike throw PST *’Wong threw away the bike.’*	王鑫 被 自行车 扔 了。 Wong.M PASS bike throw PST *’Wong was thrown away by the bike.’*
Coverb	李文清 乘 卡车 到达-了 咖啡店。 Lee.M ride truck arrive-PST coffee shop *’Lee went to the coffee shop by truck.’*	李文清 于 卡车 到达-了 咖啡店。 Lee.M at truck arrive-PST coffee shop *’Lee went to the coffee shop at truck.’*
NP head finality	王梦 正在 卖 张红梅 清洗-过-的 推车。 Wong.F PROG sell May.F clean-PRF-ADJ trolley *‘Wong is selling the trolley that Mel has cleaned.’*	王梦 正在 卖 推车 张红梅 清洗-过-的。 Wong.F PROG sell trolley May.F clean-PRF-ADJ *‘Wong is selling the trolley that Mel has cleaned.’*
Classifier	张杰 正在 穿过 一 家 艺术画廊。 Jay.M PROG pass one CL:INSTITUTION art gallery *’Jay is passing through an art gallery.’*	张杰 正在 穿过 一 段 艺术画廊。 Jay.M PROG pass one CL:LENGTH art gallery *’Jay is passing through an art gallery.’*
Filler gap	图书馆， 我 开车 去-过 这个地方。 The library, I drive to-PRF this place *‘The library, I have driven to this place.’*	图书馆， 我 开车 去-过 博物馆。 The library, I drive to-PRF the museum *‘The library, I have driven to the museum.’*
Passive	这些 患者 被 转移-了。 These patient PASS transfer-PST *’These patients were transferred.’*	这些 患者 被 下降-了。 These patient PASS fall-PST *’These patients were fell.’*
Verb complement	王慧 的 文章 吓 坏 了 包曼玉 。 Wong.F POSS article frighten badly PST Bao.F. *’Wong’s article frightened Bao badly.’*	王慧 的 文章 吓 开 了 包曼玉 。 Wong.F POSS article frighten openly PST Bao.F. *’Wong’s article frightened Bao openly.’*

Table 9: Minimal pairs from each of the nine linguistic phenomena covered by CLiMP. Differences are underlined. The second line of each example shows a gloss, and the third line is an English translation.

B.2 URLs and Licenses

Name	Paper	URL	License
BLiMP	Warstadt et al. (2020)	`https://github.com/alexwarstadt/blimp`	CC-BY
CLiMP	Xiang et al. (2021)	`https://github.com/beileixiang/CLiMP`	Not articulated

Table 10: URLs and licenses of the benchmarks.

C Computational Budgets

For each experiment on a method or a combination of a method and template, we used a single NVIDIA A6000 GPU with 48GB RAM. The total GPU hours are estimated to be about 126 hours and 21 hours for the BLiMP and CLiMP experiments, respectively.

D Results and Analysis

D.1 Max Accuracy

	Llama-3	Llama-3-Instruct	Mixtral	Mixtral-Instruct	Qwen2	Qwen2-Instruct
LP	79.6	77.1	82.5	82.3	80.4	79.7
MeanLP	77.1	74.8	79.6	79.4	77.7	77.1
PenLP	79.2	76.8	82.2	82.0	79.9	79.2
In-template LP	**85.0**	84.2	84.6	84.5	84.1	81.3
In-template MeanLP	83.1	82.6	83.2	83.1	82.8	80.5
In-template PenLP	84.4	83.5	84.5	84.3	83.6	81.2
In-template compar. LP	76.4	66.0	75.2	69.8	67.7	63.6
A/B prompting	79.5	83.9	81.9	83.6	81.5	82.8
Yes/No prob comp	77.8	**89.3**	**85.6**	**87.5**	**89.2**	**87.4**

(a) BLiMP

	Llama-3	Llama-3-Instruct	Yi-1.5	Yi-1.5-Chat	Qwen2	Qwen2-Instruct
LP	83.2	80.4	86.8	85.3	85.4	85.4
MeanLP	74.5	71.7	75.9	74.5	74.5	74.3
PenLP	80.3	77.7	84.3	82.3	82.2	82.0
In-template LP	**86.1**	**83.2**	**87.9**	**87.8**	**88.2**	**86.5**
In-template MeanLP	80.7	78.5	80.4	81.0	78.9	79.0
In-template PenLP	83.6	81.1	85.1	84.7	83.8	83.5
In-template compar. LP	72.2	61.6	68.2	67.3	74.2	63.5
A/B prompting	71.7	74.2	78.6	79.7	83.2	82.5
Yes/No prob comp	78.1	78.4	79.7	82.1	87.5	84.3

(a) CLiMP

Table 11: Percentage max accuracy by method and model. The bold font denotes the best scores. Underlines denote the second best.

D.2 Why A/B prompting does not perform well

BLiMP						CLiMP					
Llama-3	Llama-3 -Inst.	Mixtral	Mixtral -Inst.	Qwen2	Qwen2 -Inst.	Llama-3	Llama-3 -Inst.	Yi-1.5	Yi-1.5 -Chat	Qwen2	Qwen2 -Inst.
55.0	56.6	70.1	45.9	54.0	45.7	70.5	72.8	52.7	46.7	35.9	46.1

Table 12: Percentage proportion of A in the predictions of A/B prompting (averaged over templates) by model.

The low performance of A/B prompting can be partly attributed to a preference for a specific choice identifier, A or B. Table 12 shows all models except for Yi-1.5 models are at least 7 points more likely to predict one choice over the other one, even though the gold label is sampled from a uniform distribution. This suggests that the current LLMs suffer from selection bias on multiple choices as argued by Zheng et al. (2024).

D.3 Is Yes/No prob comp strong where the acceptable sentence is longer than the unacceptable?

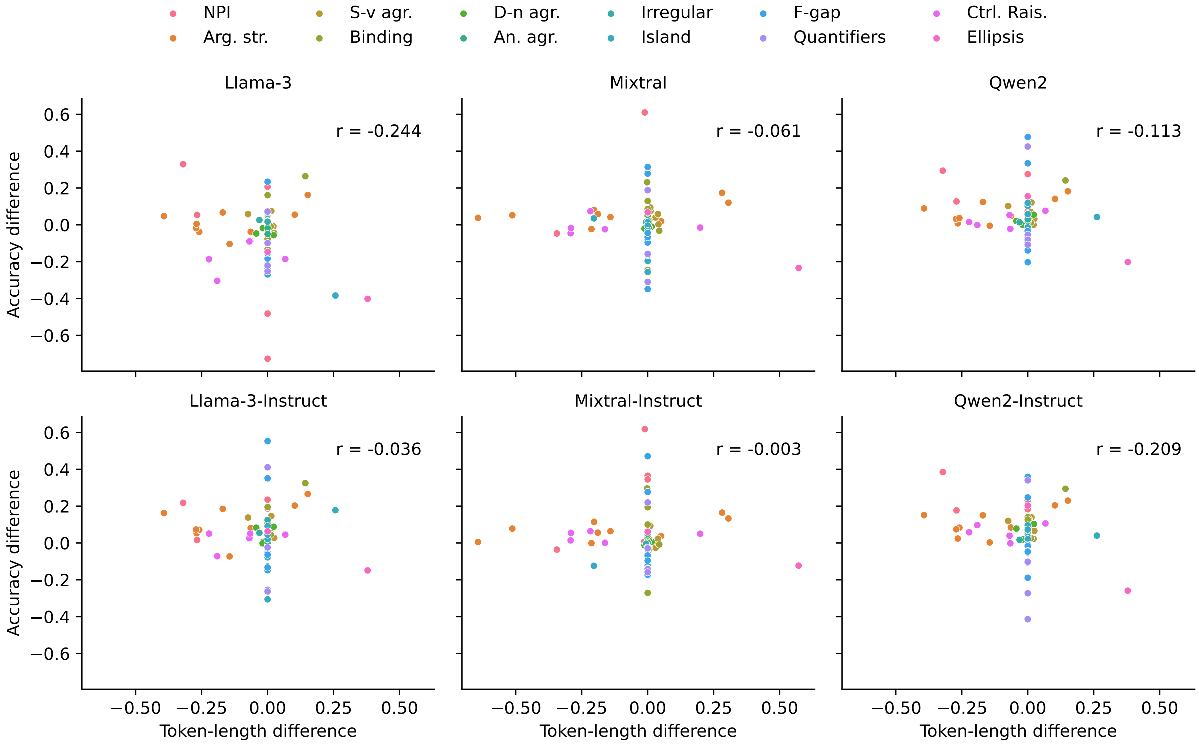


Figure 6: Correlation between the token-length difference ($|s_{\text{acceptable}}| - |s_{\text{unacceptable}}|$) and the accuracy difference (accuracy $_{\text{Yes/No prob comp}}$ $-$ accuracy $_{\text{In-template LP}}$) by model. Each dot represents a paradigm. Plots are annotated with the Pearson correlation coefficient r.

Figure 6 shows that Yes/No prob comp is not particularly accurate compared to in-template LP in phenomena where the acceptable sentence is, on average, longer than the unacceptable one. We only find no or weak negative correlations between the accuracy difference and token-length difference.

D.4 Voting Ensembles

	Llama-3	Llama-3-Instruct	Mixtral	Mixtral-Instruct	Qwen2	Qwen2-Instruct
Ensemble P-only	76.0	89.0	85.4	84.4	89.5	87.0
Ensemble Mix-P3	82.7	**89.7**	**87.5**	**86.6**	**90.2**	**87.7**
Ensemble Mix-L3	**86.1**	86.3	86.5	86.4	86.7	84.8
Ensemble L-only	85.1	84.1	84.6	84.3	84.3	81.0
In-template LP (oracle)	85.0	84.2	84.6	84.5	84.1	81.3
Yes/No prob comp (oracle)	77.8	89.3	85.6	87.5	89.2	87.4

(a) BLiMP

	Llama-3	Llama-3-Instruct	Yi-1.5	Yi-1.5-Chat	Qwen2	Qwen2-Instruct
Ensemble P-only	76.0	76.6	78.4	81.9	87.6	84.1
Ensemble Mix-P3	79.1	78.4	79.9	83.9	89.4	87.1
Ensemble Mix-L3	**87.1**	**83.8**	**88.7**	**88.9**	**90.5**	**89.4**
Ensemble L-only	86.1	83.0	88.0	87.7	88.4	87.0
In-template LP (oracle)	86.1	83.2	87.9	87.8	88.2	86.5
Yes/No prob comp (oracle)	78.1	78.4	79.7	82.1	87.5	84.3

(b) CLiMP

Table 13: Percentage accuracy of voting ensembles of in-template LP and Yes/No prob comp, with the oracle (max) accuracy by single methods (best template). The bold font denotes the best ensemble score. Underlines denote oracle results surpassing the best ensemble result.