

# Read it in Two Steps: Translating Extremely Low-Resource Languages with Code-Augmented Grammar Books

Chen Zhang\*, Jiuheng Lin\*, Xiao Liu, Zekai Zhang, Yansong Feng†

Wangxuan Institute of Computer Technology, Peking University

{zhangch, fengyansong}@pku.edu.cn

linjiuheng@stu.pku.edu.cn

## Abstract

While large language models (LLMs) have shown promise in translating extremely low-resource languages using resources like dictionaries, the effectiveness of grammar books remains debated. This paper investigates the role of grammar books in translating extremely low-resource languages by decomposing it into two key steps: grammar rule retrieval and application. To facilitate the study, we introduce ZHUANGRULES, a modularized dataset of grammar rules and their corresponding test sentences. Our analysis reveals that rule retrieval constitutes a primary bottleneck in grammar-based translation. Moreover, although LLMs can apply simple rules for translation when explicitly provided, they encounter difficulties in handling more complex rules. To address these challenges, we propose to represent grammar rules as code functions, motivated by their similarities in structures and the benefit of code in facilitating LLM reasoning. Our experiments show that using code rules significantly boosts both rule retrieval and application, ultimately resulting in a 13.1% BLEU improvement in translation.

## 1 Introduction

Most human languages suffer from data scarcity (Joshi et al., 2020). With only a few thousand sentences available for extremely low resource (XLR) languages, traditional pretraining or finetuning methods (Yong et al., 2023; Liu et al., 2020) are impractical for building effective machine translation (MT) systems. Facing the challenge of XLR MT, large language models (LLMs) offer a promising alternative. Recent research reveals that LLMs can perform XLR MT through in-context learning (ICL), leveraging small-scale linguistic resources like dictionaries and parallel sentences (Tanzer et al., 2024; Zhang

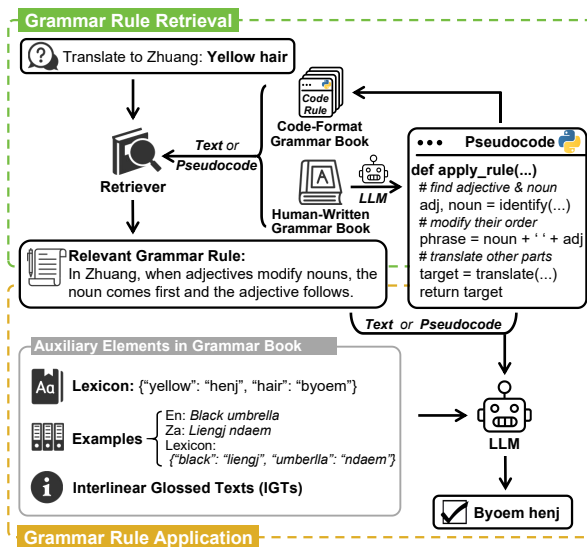


Figure 1: An illustration of LLMs using grammar rules in code format to translate Zhuang, an extremely low-resource language.

et al., 2024a). Among these resources, grammar books, with their systematic linguistic descriptions, appear ideal for guiding translation, but their effectiveness remains debated. Some studies claim that prompting LLMs with full grammar books improves translation performance (Tanzer et al., 2024; Gemini Team et al., 2024; Zhang et al., 2024b), while others argue that such improvements may stem from lexical leakage, where LLMs identify the bilingual explanations of several words in the test sentence from the grammar book and use them as shortcuts, rather than genuinely understanding grammar rules (Aycock et al., 2024). However, no existing dataset effectively eliminates such interference factors, making it difficult to assess whether LLMs truly understand grammar rules.

To address this gap, we introduce ZHUANGRULES, a fine-grained dataset focusing on the grammar rule understanding for Zhuang (ISO 639-1: za), a language frequently studied in XLR

\*Equal contribution.

†Corresponding author.

MT (Zhang et al., 2024a; Bai et al., 2024). ZHUANGRULES decomposes grammar books into modular elements, consisting of 109 atomic grammar rules, each paired with an average of 5.6 Zhuang-Chinese parallel sentences for testing. We ensure that correctly translating each test instance requires applying its corresponding grammar rule. We provide a Zhuang-Chinese lexicon for each sentence during testing, to disentangle grammar rule comprehension from lexical knowledge. These designs enable more controlled and interpretable evaluation in LLMs’ ability of grammar understanding.

Given that each test sentence in ZHUANGRULES is annotated with its relevant grammar rule, we conduct a pilot study comparing two settings: providing all grammar rules (akin to an entire grammar book) versus supplying only the necessary one. We find that the latter significantly outperforms the former, suggesting that grammar-based MT relies heavily on LLMs’ ability to identify the required rules. We thus break down grammar-based MT into two stages, grammar rule retrieval and grammar rule application, and explore the following research questions: (1) **RQ1**: Can LLMs retrieve the grammar rules required for translating a test sentence? (2) **RQ2**: Can LLMs effectively apply a given rule for translation as instructed?

We discover that grammar rule retrieval is a significant bottleneck in XLR MT, as LLMs struggle to effectively locate the necessary rules. Regarding rule application, we find LLMs can apply simple rules for translation when explicitly provided, with further gains observed when auxiliary elements like parallel sentence examples and interlinear glossed texts (IGTs) are included in the prompt. However, handling complex rules involving multiple actions remains a challenge for LLMs, with performance dropping to half that of simpler cases.

We further explore strategies to enhance LLMs’ ability to utilize grammar rules in these two steps. Inspired by prior works on improving LLM reasoning through code representations (Liu et al., 2023; Li et al., 2024), we observe a strong analogy between the sequential operations in grammar rule application and the procedural structures of code. For instance, adding affixes to a word resembles an arithmetic addition operation, while selecting different affixes based on conditions aligns with an if-else structure in code. Therefore, as illustrated in Figure 1, we convert grammar rules into code-based representations using GPT-4o (OpenAI et al., 2024), to facilitate LLMs in translation. Ad-

ditionally, we propose RULE-BY-RULE retrieval, a simple but effective strategy that examines the necessity of each rule individually instead of processing the whole book directly.

Our experiments show that retrieving grammar rules in code format improves recall by 8.8% compared to textual rules, and enhances LLMs’ ability to utilize given rules effectively, boosting the translation performance by 12.2% BLEU on ZHUANGRULES. This benefit is also observed on MTOB (Tanzer et al., 2024), another translation benchmark for XLR MT. Finally, combining code rule with RULE-BY-RULE retrieval strategy outperforms the end-to-end translation using the textual grammar book by 13.1% BLEU on ZHUANGRULES.

Our contributions are summarized as follows: (1) We underscore the necessity of breaking down grammar-based MT into two steps, rule retrieval and application, and identify rule retrieval as a major bottleneck. (2) We introduce a code-based format for grammar rules, improving LLMs’ abilities in both steps and yielding substantial gains in translation performance. (3) We present ZHUANGRULES, a dataset for explainable research on XLR MT using grammar rules, decomposing grammar books into structured elements including rules, parallel sentences, lexicons, and IGTs. Our data and code are publicly available to the community<sup>1</sup>.

## 2 Dataset: ZHUANGRULES

We study the problem of grammar understanding using Zhuang, a low-resource language in China, which current LLMs hardly understand (Zhang et al., 2024a). We collect ZHUANGRULES, a set of 109 rules on Zhuang grammar written in Chinese. Each rule is paired with several Zhuang-Chinese parallel phrases/sentences for testing, amounting to 608 pairs. Each pair is further annotated with a bilingual lexicon covering all relevant lexical items in the sentences, which can disentangle the interference from LLMs’ lack of Zhuang lexical knowledge when evaluating their understanding of grammar rules. Compared to previous resources of complete grammar books (Tanzer et al., 2024; Zhang et al., 2024b; Hus and Anastasopoulos, 2024), ZHUANGRULES enables more systematic and controllable analysis with its modularized structures for XLR MT.

<sup>1</sup><https://github.com/Infinite-set/ZhuangRules>

## 2.1 Rule Collection

We collect the rules from two books on Zhuang written in Chinese, 《壮语通论》 (*General Introduction to Zhuang Language*; Wei and Qin, 2006) and 《壮语基础教程》 (*Basic Course of Zhuang Language*; Wei and Qin, 2008). From these books, we collect grammar rules of Zhuang and their Zhuang-Chinese parallel phrases/sentences, which are typically concise and concretely illustrate the usage of the corresponding rule. For each sentence, we provide a Zhuang-Chinese lexicon covering the words appearing in the Zhuang sentence, which helps eliminate the interference to the experiments caused by the model not knowing word meanings. The following is a grammar rule and one of its parallel examples. See data statistics and details of collection in Appendix B.

**Rule:** 在壮语中, 形容词作名词的修饰时, 名词在前, 形容词于后。(In Zhuang, when adjectives modify nouns, the noun comes first and the adjective follows.)

**Example:**

Zhuang: byoem henj

Chinese: 黄头发 (*yellow hair*)

Lexicon: {byoem: 头发(*hair*), henj: 黄(*yellow*)}

## 2.2 Analysis of Rules

Zhuang exhibits diverse linguistic features in its grammar rules. To gain a deeper understanding of these rules, we annotate each with fine-grained attributes including **action**, **difficulty**, and **domain**. These attributes provide clear categorization, enabling detailed analysis of their utilization in XLR MT.

Regarding **action**, we identify the atomic operations required in applying each grammar rule for Chinese-Zhuang translation, such as adding affixes and reordering two words. The number of actions in a grammar rule can reflect its difficulty. We evaluate the **difficulty** of each rule based on the number of involved actions and the degree of difference between Zhuang and Chinese. The rules are thereby categorized into three levels: *easy*, *medium*, and *hard*, with the average number of required operations being 1.2, 1.5 and 2.1, respectively. Additionally, we label rules according to their linguistic **domain**, following the taxonomy in WALS (Dryer and Haspelmath, 2013). We find that most rules in ZHUANGRULES deal with morphology and word order. See Appendix B.4 for detailed categorization of each attribute.

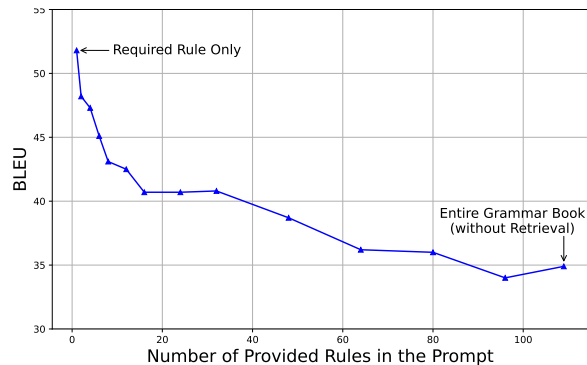


Figure 2: Change in translation performance of providing varying numbers of irrelevant grammar rules in addition to the required rule (Qwen-2.5-72B-Instruct, Chinese  $\rightarrow$  Zhuang).

## 3 Methodology

Motivated by a pilot study that highlights the significant impact of the number of provided grammar rules on MT performance, we formalize the translation process using grammar books as a two-step procedure: grammar rule retrieval and grammar rule application. To gain deeper insights into LLMs’ capabilities, we evaluate their performance at each stage independently. Additionally, we explore strategies to enhance LLMs’ abilities in these two stages, especially representing grammar rules in a structured code-based format.

### 3.1 Pilot Study: Providing LLMs with Varying Numbers of Rules

Using ZHUANGRULES, we conduct a pilot study to assess whether providing varying numbers of grammar rules in the prompt affects LLMs’ abilities of utilizing them for translation.

As shown in Figure 2, we observe that the translation performance declines sharply when we begin with only the required rule and progressively add more irrelevant rules<sup>2</sup>. This suggests that LLMs struggle with numerous irrelevant rules in the grammar book, highlighting that the performance of grammar-based XLR MT is closely related to LLMs’ ability to identify the required grammar rules.

Based on these findings, we are motivated to decouple rule retrieval from the process of end-to-end grammar-based translation, and evaluate two separate abilities of LLMs: finding the relevant grammar rules from a grammar book, and applying

<sup>2</sup>We observe similar trends in both Chinese-to-Zhuang and Zhuang-to-Chinese translation. See details in Appendix D.1.

them in translation.

### 3.2 Retrieving Grammar Rules

Our pilot study indicates that the performance of grammar-based XLR MT is closely related to LLMs’ ability to identify the required grammar rules. Therefore, given a sentence to be translated, we first explore whether LLMs can find the required rules from a grammar book, which is simulated by the concatenation of all the rules in ZHUANGRULES (approximately 4K tokens).

We mainly explore whether changing the task format or presentation of grammar rules affects LLMs’ abilities of finding required grammar rules. As a baseline, we adopt **BM25** (Robertson et al., 2009), where we use the test sentence as the query and retrieve the top  $k$  relevant rules from the grammar book.

**Changing the Task Format** Similar to the default setting of performing XLR translation with a full book in previous works (Tanzer et al., 2024), given a grammar book (a concatenation of rules) and a sentence to be translated as input, we instruct LLMs to output relevant rules in the book. We refer to this strategy as **FULL-BOOK**.

Considering the **FULL-BOOK** approach places a high demand on the model’s ability of long-context understanding, we propose another **RULE-BY-RULE** strategy with a much shorter input length. In this approach, instead of handling the entire grammar book at once, LLMs examine whether each rule in the grammar book is relevant to the sentence to be translated individually. Specifically, we input one candidate grammar rule and a test sentence at a time, and require LLMs to perform binary classification over their relevance.

**Changing the Representation of Rules** Previous works find that code formats can enhance the reasoning capabilities of LLMs in tasks involving math or logic (Chen et al., 2023; Liu et al., 2023). These advantages stem from the similarities between these tasks and code representations in terms of the modular structure and control flow. Similarly, operations involved in grammar rules exhibit a natural resemblance to code. For example, adding or removing affixes resembles arithmetic operations, while selecting different affixes based on conditions can be represented with an `if-else` structure in code. Building on this observation, we investigate whether the benefits of code-based reasoning extend to grammar rule understanding.

We convert the textual rules in ZHUANGRULES into **code rules** with GPT-4o (OpenAI et al., 2024). Each code rule consists of two parts: (1) a concise comment outlining the steps for applying the rule in translation, and (2) a pseudo-code function that simulates the translation process. We only provide 5 exemplars for ICL during conversion, without applying further constraints on the structure or style of the pseudo-code since LLMs exhibit robustness to variations in code style and format when reasoning with code (Liu et al., 2023). We randomly sample 10 code rules for quality check and find GPT-4o is effective at generating pseudocodes. All samples follow proper Python syntax, and only one sample omits minor information of the original rule. The remaining samples are all complete and accurate in both translation directions. See examples of code rules in Appendix C.

### 3.3 Applying Grammar Rules

After exploring the retrieval stage in XLR MT, we investigate LLM’s ability to apply specific grammar rules in a targeted manner, by asking LLMs to translate a sentence directly using the required rule. We explore whether LLMs exhibit preferences for different representations of grammar rules and further examine the impact of auxiliary components commonly included in grammar books, such as parallel examples and IGTs.

**Presentation of Rules** We mainly investigate the effects of rule formats on grammar rule application, by comparing the settings using textual and code rules. We additionally examine how the language used for writing rules affects translation, which is discussed in Appendix D.2.

**Auxiliary Elements** We further explore whether providing auxiliary elements from grammar books in the prompt can facilitate LLMs to apply rules.

**Parallel examples** can demonstrate the use of grammar rules more intuitively and are considered effective in XLR translation (Court and El-sner, 2024; Aycock et al., 2024). In addition to the required rule, we provide each test sentence with several pairs of parallel sentences requiring the same rule.

**Interlinear glossed text** (IGT) is a line-by-line format for annotating linguistic corpora, where each morpheme is labeled with a descriptive annotation. It has been widely adopted as an intermediate form for low-resource language translation (Ginn et al., 2024a,b; Ramos et al., 2024).

As the two grammar books used for collecting ZHUANGRULES do not provide IGTs for the parallel sentences in them, we use GPT-4o to generate IGT annotations for each Zhuang sentence in ZHUANGRULES, with 123 IGTs collected from 《壮语语法标注文本》 (*Annotated Zhuang Grammar Text*; Lan, 2016) as ICL exemplars. We also conduct a quality check, finding that GPT-4o generates the correct symbols for 72% of the morphemes. See details in B.2.

In XLR translation experiments, we follow the approach described in Ramos et al. (2024), where each parallel sentence in the prompt is paired with its corresponding IGT, and LLMs are instructed to first generate the IGT for the test sentence before translating.

## 4 Experiments and Analyses

We answer our two research questions by analyzing the current capabilities of LLMs and exploring the benefits of representing rules in a code-based format. We then aggregate our findings obtained from the two-step investigation to summarize the best practice for using grammar books in XLR MT.

### 4.1 Experimental Setups

**Models** We use three open-source LLMs for experiments: Qwen-2.5-7B-Instruct, Qwen-2.5-72B-Instruct (Yang et al., 2024), and Llama-3.1-70B-Instruct (Dubey et al., 2024).

**Datasets** ZHUANGRULES is the only dataset so far supporting controllable and interpretable experiments on grammar books. To validate the generalizability of our conclusions, we additionally use MTOB (Tanzer et al., 2024) for experiments, which contains a grammar book for Kalamang (ISO 639-3: kgv), an XLR language in Indonesia. We extract paragraphs of grammar descriptions and their corresponding examples from the book using regular expressions<sup>3</sup>. See details of data construction from MTOB in Appendix B.3.

**Metrics** For BM25, we report recall@ $k$ , reflecting whether the relevant rule appears in the top- $k$  retrieval results. For FULL-BOOK and RULE-BY-RULE retrieval with LLMs, as the number of retrieved rules is not pre-defined, we report the aver-

<sup>3</sup>The MTOB grammar book is written in a progressive style, with most example sentences requiring multiple rules for accurate translation. As we are not able to annotate all the required rules, we exclude MTOB from rule retrieval experiments and only use it for rule application experiments.

	za2zh		zh2za	
<b>BASELINE</b>	<b>rec@1</b>	<b>rec@5</b>	<b>rec@1</b>	<b>rec@5</b>
BM25	26.3	41.6	13.5	27.3
<b>FULL-BOOK</b>	<b>rec</b>	<b>#rules</b>	<b>rec</b>	<b>#rules</b>
Qwen-2.5-7B	3.1	2.3	5.2	2.2
Llama-3.1-70B	33.9	3.2	22.9	2.5
Qwen-2.5-72B	52.8	1.8	49.4	1.8
<b>RULE-BY-RULE</b>	<b>rec</b>	<b>#rules</b>	<b>rec</b>	<b>#rules</b>
Qwen-2.5-7B (text)	55.1	2.5	67.9	4.0
Qwen-2.5-7B (code)	68.4	3.8	80.3	4.7
Llama-3.1-70B (text)	69.7	2.2	75.8	3.6
Llama-3.1-70B (code)	82.2	4.2	<b>87.5</b>	5.5
Qwen-2.5-72B (text)	<u>89.4</u>	4.1	84.7	4.4
Qwen-2.5-72B (code)	<b>89.6</b>	3.9	<u>87.1</u>	4.1

Table 1: Performance of different rule retrieval strategies on ZHUANGRULES. The best scores are made **bold**, with the second underlined.

age number and recall of retrieved rules. For rule application, we report BLEU (Papineni et al., 2002) and chrF++ (Popović, 2017), using the implementation from Post (2018).

**Prompting** For the experiments involving parallel sentence examples in grammar rule application, we randomly sample two pairs of examples for ICL from those requiring the same rule with the test instances. For all settings except the ones explicitly annotated with **w/o Lexicon**, we include Zhuang-Chinese lexicons covering the words appearing in the prompt by default. See all the prompts used in our experiments in Appendix C.

### 4.2 RQ1: Can LLMs Find the Required Grammar Rule?

**LLMs struggle to retrieve rules from the full grammar book directly.** In Table 1, we compare the performance of different strategies for finding relevant rules for the sentence to be translated. BM25, relying solely on lexical overlap, fails to retrieve the relevant rules for more than half of the test instances within the top-5 results, underscoring the complexity of rule retrieval.

Using LLMs to identify the required rules from the entire grammar book (FULL-BOOK) shows notable improvements over BM25, as LLMs can better capture the semantic relationships between rules and testing instances beyond simple lexical matching. However, their performance is still far from perfect, with the best results still hovering around 50% in recall. Moreover, this approach is highly dependent on model capabilities, as weaker models,

BLEU / chrF++	Zhuang → Chinese			Chinese → Zhuang			Average
	Qwen (72B)	Llama (70B)	Qwen (7B)	Qwen (72B)	Llama (70B)	Qwen (7B)	
<i>Baselines</i>							
No Rule (w/o Lexicon)	2.7 / 1.9	0.3 / 0.3	1.0 / 0.8	0.7 / 8.1	0.6 / 4.8	0.1 / 2.2	0.9 / 3.0
No Rule	31.2 / 28.3	28.7 / 27.0	25.7 / 24.0	22.1 / 49.1	21.4 / 49.9	24.1 / 49.5	25.5 / 38.0
Parallel Examples	65.6 / 61.3	58.0 / 55.6	54.7 / 50.7	63.4 / 80.1	61.8 / 80.6	57.4 / 75.8	60.2 / 67.4
+ Synthetic IGT	65.6 / 62.6	60.4 / 60.3	56.3 / 55.7	- / -	- / -	- / -	- / -
<i>Textual Grammar Rules</i>							
Random Textual Rule	31.8 / 29.0	28.4 / 27.8	24.8 / 23.9	21.6 / 49.3	18.7 / 42.8	15.7 / 43.6	23.5 / 36.1
Gold Textual Rule	51.4 / 50.5	47.8 / 48.7	39.4 / 40.6	51.8 / 78.7	47.9 / 79.3	35.6 / 66.2	45.7 / 60.7
+ Parallel Examples	<u>70.7 / 68.0</u>	<u>68.4 / 64.9</u>	<u>58.5 / 57.3</u>	<u>80.7 / 91.0</u>	<b>78.9 / 90.4</b>	<u>63.9 / 80.6</u>	<u>70.2 / 75.4</u>
+ Synthetic IGT	67.9 / 66.6	62.2 / 64.0	<u>60.0 / 58.7</u>	- / -	- / -	- / -	- / -
<i>Code Grammar Rules</i>							
Random Code Rule	27.7 / 25.9	23.7 / 25.2	22.3 / 22.8	20.2 / 48.7	18.4 / 48.0	15.2 / 44.0	21.3 / 35.8
Gold Code Rule	63.3 / 61.1	57.8 / 57.9	49.5 / 50.0	69.3 / 86.7	55.2 / 83.1	52.0 / 76.1	57.9 / 69.2
+ Parallel Examples	<b>73.4 / 71.4</b>	<b>72.0 / 69.9</b>	<b>62.3 / 61.0</b>	<b>81.2 / 91.6</b>	<u>77.8 / 90.1</u>	<b>67.7 / 83.4</b>	<b>72.4 / 77.9</b>
+ Synthetic IGT	68.5 / 66.8	65.1 / 64.4	59.8 / <u>59.5</u>	- / -	- / -	- / -	- / -

Table 2: Translation performance of different settings of rule application on ZHUANGRULES. Note that IGTs do not support high-to-low-resource language translation, i.e. Chinese → Zhuang. The best scores are made **bold**, with the second underlined.

	kgv2eng		eng2kgv	
	BLEU	chrF++	BLEU	chrF++
No Rule (w/o Lex.)	1.6	9.3	0.8	8.6
No Rule	12.0	34.6	39.9	63.8
Random Textual Rule	11.9	34.8	40.7	63.5
Random Code Rule	13.3	36.3	37.9	61.9
Gold Textual Rule	14.6	39.2	43.8	<b>67.3</b>
Gold Code Rule	<b>16.0</b>	<b>40.7</b>	<b>44.5</b>	67.0

Table 3: Translation performance of Qwen-2.5-72B-Instruct between English and Kalamang, using different rule formats.

such as Qwen-2.5-7B-Instruct, are almost unable to find the correct rules.

This finding indicates that when LLMs perform end-to-end translation with a grammar book, they mostly do not know which grammar rules are necessary for translating a given sentence, in line with our pilot study in Figure 2, where we find a strong correlation between rule retrieval performance and final translation quality. By addressing the bottleneck of rule retrieval, it is possible to further improve the translation performance of LLMs. We then explore two strategies for improving rule retrieval: transforming the task format and changing the form of rules.

**Converting the retrieval task into RULE-BY-RULE classification helps.** As shown in Table 1, instead of providing the entire grammar book (FULL-BOOK), examining each rule individually (RULE-BY-RULE) leads to significantly better

performance. This approach achieves nearly 80% recall with fewer than 5 retrieved rules on average, making it a more practical solution compared to providing LLMs with the entire book (FULL-BOOK).

**Rules in code format are more LLM-friendly for retrieval.** As shown in Table 1, converting textual rules into code forms, combined with the RULE-BY-RULE strategy, improves retrieval performance across all models, achieving up to 90% recall while maintaining a manageable number of retrieved rules. Code forms transform descriptive rules into procedural knowledge, making it easier for LLMs to understand the requirements of translation than textual rules.

### 4.3 RQ2: Can LLMs Apply a Given Grammar Rule as Instructed?

**LLMs can apply simple rules for translation when explicitly given.** As shown in Table 2, providing the necessary textual grammar rule significantly improves translation performance over providing no or random grammar rules. For example, on ZHUANGRULES we observe an absolute increase of 26% chrF++ in Qwen-2.5-72B’s performance, after providing the gold grammar rules. This finding is also validated in another XLR language, Kalamang. As shown in Table 3, there is an increase of 4% chrF++ on Kalamang-English translation after providing relevant rules, compared to the settings providing no or random rules. These

improvements can be attributed to LLMs’ strong capability to comprehend and follow clearly-given instructions.

We then investigate the role of other auxiliary elements of the grammar books in facilitating the understanding of grammar rules. In the absence of explicit grammar rules, parallel examples and IGTs prove valuable for translation, aligning with findings from previous work (Aycock et al., 2024; Ginn et al., 2024a). After pairing textual grammar rules with parallel examples, we observe an average chrF++ gain of 14.7% over using rules alone. We conjecture that parallel examples help LLMs identify common patterns or usages in a more intuitive way than rules alone. Besides, their format resembles testing instances, aiding LLMs in better mimicking the translation process. However, further incorporating synthetic IGTs when grammar rules are provided reduces the gain from parallel examples. This may be due to the noise introduced by the GPT-4o generated IGTs (see Appendix B.2), which may be inconsistent with the provided rules.

**Code formats enhance rule understanding, especially for difficult rules.** As shown in Table 2, code rules consistently outperform textual rules, resulting in an average improvement of 8.5% chrF++ on ZHUANGRULES. This trend holds for Kalamang as well, as seen in Table 3, where code rules also outperform textual rules in most settings.

The advantage of code format is more pronounced on difficult grammar rules, which involve more operation steps when applied for translation. As shown in Table 4, the improvement of code rules over textual ones is particularly noticeable on the *hard* subset of ZHUANGRULES, especially when parallel examples are unavailable. See Appendix E for a case study.

Beyond performance improvements, code formats exhibit more advantages as a structural representation for XLR translation. It offers a language-agnostic interface for understanding grammar, without requiring pre-defined language-specific protocols such as the glossing conventions required by IGTs (Rules, 2008). Furthermore, the conversion into code format can be well performed by powerful LLMs, requiring little human labor while yielding substantial performance gains.

**Code formats support simultaneous application of multiple rules.** In the previous experiments, we adopt an idealized setting where translating a testing sentence requires only a single grammar rule,

	za2zh			zh2za		
	<i>easy</i>	<i>medium</i>	<i>hard</i>	<i>easy</i>	<i>medium</i>	<i>hard</i>
No Rule	29.5	28.8	24.2	18.6	22.2	30.5
Text Rule	65.6	51.3	34.6	85.5	82.4	69.3
Code Rule	76.3	57.9	48.6	93.0	87.5	76.8
$\Delta$	+10.7	+6.6	<b>+14.0</b>	<b>+7.5</b>	+5.2	<b>+7.5</b>
Text Rule + 2 Ex.	78.8	69.1	52.0	95.2	91.1	85.6
Code Rule + 2 Ex.	81.1	72.6	56.7	94.9	92.0	87.2
$\Delta$	+2.3	+3.5	<b>+4.6</b>	+0.3	+1.0	<b>+1.6</b>

Table 4: Translation performance (chrF++) of Qwen-2.5-72B-Instruct on ZHUANGRULES in different levels of difficulty. We additionally report the performance difference ( $\Delta$ ) between using textual and code rules.

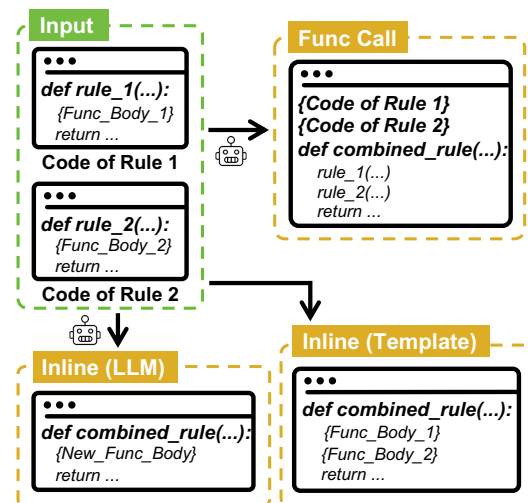


Figure 3: An illustration of three strategies for combining multiple rules.

to facilitate controlled experiments. However, in real-world scenarios, translating a sentence often necessitates the application of multiple grammar rules. To validate our findings under a more realistic setting, we collect a small set of 96 testing instances requiring multiple rules. Concretely, we select suitable candidates from the parallel sentences in ZHUANGRULES and modify them to require two rules for translation. See Appendix B for details.

Using code formats for multiple grammar rules in translation is non-trivial. The structural nature of code enables diverse strategies for combining these rules. As shown in Figure 3, we explore three combining strategies. In the FUNC CALL strategy, LLMs generate a new function that determines the order of rule application and calls two existing rule functions sequentially. For the other two INLINE strategies, the bodies of two rules compose the new function. In INLINE (Template), we directly concatenate the bodies of two rules in random or-

	za2zh		zh2za	
	BLEU	chrF++	BLEU	chrF++
No Rule	30.2	25.2	20.9	47.9
One Textual Rule	39.5	31.6	39.5	64.0
Two Textual Rules	47.3	41.7	45.5	73.9
FUNC CALL	46.6	44.7	50.6	72.3
INLINE (Template)	56.0	48.5	<b>51.4</b>	<b>75.5</b>
INLINE (LLM)	<b>56.7</b>	<b>52.7</b>	50.4	74.7

Table 5: Translation performance of Qwen-2.5-72B using different strategies for utilizing multiple grammar rules.

der into a new function, while in INLINE (LLM), LLMs freely generate a new function combining the contents of two rule functions.

As shown in Table 5, providing all necessary rules significantly improves translation performance compared to providing no or one rule, demonstrating that LLMs can effectively apply multiple rules. Consistent with the findings in single-rule application, code rules yield noticeable advantages over textual ones. Among the strategies for combining multiple code rules, INLINE (LLM) achieves the best overall performance, indicating that LLMs can find a more effective way of organizing multiple rules by themselves.

#### 4.4 Putting it Together: Best Practice of Utilizing Grammar Rules

In previous experiments, we decompose the grammar-based translation process into two steps and examine them independently. We find that although LLMs show promising abilities to apply it for translation when given a relevant grammar rule, they struggle to find relevant rules from a long grammar book for a sentence to be translated. To mitigate the bottleneck of rule retrieval and enhance the application of harder rules, we propose a RULE-BY-RULE retrieval strategy and represent rules in code format. These approaches enhance the performance of both steps individually.

Now we present a comprehensive evaluation of the entire pipeline, assessing various combinations of design choices for each step. As shown in Table 6, the gains in rule retrieval brought by RULE-BY-RULE retrieval strategy and code-based rule representations can propagate to the final translation performance, with up to 13.1% increase in chrF++ for Qwen-2.5-72B-Instruct compared to feeding the whole grammar books to LLMs, the default practice of previous works. Based on these findings, we summarize a promising strategy of

Application	Text Rule		Code Rule	
	za2zh	zh2za	za2zh	zh2za
w/o Retrieval	42.6	64.2	-	-
FULL-BOOK	43.1	66.5	50.2	69.4
RULE-BY-RULE (text)	46.8	69.9	55.4	72.3
RULE-BY-RULE (code)	47.6	71.4	<b>55.7</b>	<b>74.3</b>

Table 6: Translation performance (chrF++) of Qwen-2.5-72B-Instruct in different settings of retrieving and applying rules. The result of *w/o Retrieval + Code Rule* is not reported because the input exceeds the maximum context length of evaluated LLMs.

using grammar books with a moderate number of grammar rules: **first, converting the rules in the grammar book into code format, then employing RULE-BY-RULE retrieval with code rules, and finally applying the retrieved code rules for translation.**

## 5 Related Works

**XLR MT with LLMs** LLMs have been widely adopted for XLR MT for their training-free nature and efficient use of limited data. Existing works focusing on improving XLR MT performance using linguistic resources, such as dictionaries (Ghazvininejad et al., 2023; Elsner and Needle, 2023; Lu et al., 2024; Dimakis et al., 2024), parallel corpora (Court and Elsner, 2024; Guo et al., 2024; Liu et al., 2024; Zhang et al., 2024a), grammar rules (Tanzer et al., 2024; Gemini Team et al., 2024; Zhang et al., 2024b; Hus and Anastasopoulos, 2024; Aycock et al., 2024), and IGTs (Ginn et al., 2024a,b; Ramos et al., 2024). However, these works arrive at divergent conclusions, due to inconsistent experimental setups. In contrast, we conduct controlled experiments using decomposable data, unveiling LLMs’ true capabilities in XLR MT.

**Reasoning with Code** Code has been proven a better interface for LLMs to perform reasoning than natural languages under many circumstances such as math reasoning (Li et al., 2024) and causal reasoning (Liu et al., 2023). Existing works either use the modularity feature of code to describe structures (Madaan et al., 2022; Bogin et al., 2024) or use the control flow of code to describe logical flows (Liu et al., 2025; Puerto et al., 2024; Chae et al., 2024). We take advantage of both aspects of code to represent grammar rules, which greatly facilitate LLMs in XLR MT.

In the area of machine translation, rule-based machine translation systems, such as Apertium (For-



cada et al., 2011), employ structured rule specification languages. Our work on using code for machine translation extends this idea, adapting the concept of structured rules within the context of LLMs.

## 6 Conclusion

We present a comprehensive evaluation of LLMs’ ability to leverage grammar books for translating XLR languages. Using ZHUANGRULES, a newly introduced dataset for interpretable research, we decompose the translation process into two key steps: grammar rule retrieval and grammar rule application. Our analysis reveals that while LLMs can effectively apply simple grammar rules, they struggle significantly with retrieving the relevant rules. To address this, we propose to represent grammar rules in code format, which improves the performance of both retrieval and application, eventually enhancing the overall translation quality.

## Limitations

**Scope of Studied Languages** Collecting suitable data for controlled experiments on extremely low-resource languages requires significant effort. We do our best to investigate the research question on two XLR languages, Zhuang and Kalamang, which are considered unseen by current LLMs and thus frequently used for the research of on-the-fly XLR MT with linguistic resources (Tanzer et al., 2024; Aycock et al., 2024; Zhang et al., 2024a; Bai et al., 2024). Moreover, the substantial typological differences between Zhuang (morphologically analytic, SVO) and Kalamang (morphologically synthetic, SOV) lend credence to the generalizability of our findings.

Unlike other low-resource languages that LLMs have some preliminary support for, with a sufficiently large corpus available for fine-tuning, we believe that these extremely low-resource languages could benefit more from grammar book-based approaches. Thus, we select Zhuang and Kalamang as our primary research targets. We encourage future work to explore whether our findings generalize to low-resource languages with slightly more data availability.

**Token Efficiency** While code rules substantially enhance LLMs’ understanding of grammar, they are longer than textual rules, resulting in a higher token count in the input. However, this issue can be mitigated through effective rule retrieval over

the grammar book. As demonstrated in our experiments, the best retrieval strategy not only improves translation performance but also maintains a manageable input length.

**Idealized Scenarios** In most of our experiments, test instances are designed to require only a single grammar rule for translation. This controlled setting allows for a more precise and interpretable diagnosis of LLMs’ behavior. However, this represents an idealized scenario, as real-world translations often require applying multiple rules from grammar books. To bridge this gap, we have conducted preliminary experiments on a subset of instances requiring two rules, further validating our findings. We leave the exploration of more open-ended and unconstrained translation settings for future work.

## Acknowledgements

This work is supported in part by NSFC (62161160339) and Beijing Science and Technology Program (Z231100007423011). We thank the anonymous reviewers for their valuable feedback. We also thank Shaodan Sui for the efforts in the dataset construction. For any correspondence, please contact Yansong Feng.

## References

- Seth Aycock, David Stap, Di Wu, Christof Monz, and Khalil Sima’an. 2024. Can llms really learn to translate a low-resource language from one grammar book? *arXiv preprint arXiv:2409.19151*.
- Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, et al. 2024. Longbench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. *arXiv preprint arXiv:2412.15204*.
- Ben Bogin, Shivanshu Gupta, Peter Clark, and Ashish Sabharwal. 2024. [Leveraging code to improve in-context learning for semantic parsing](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4971–5012, Mexico City, Mexico. Association for Computational Linguistics.
- Hyungjoo Chae, Yeonghyeon Kim, Seungone Kim, Kai Tzu-iunn Ong, Beong-woo Kwak, Moohyeon Kim, Sunghwan Kim, Taeyoon Kwon, Jiwan Chung, Youngjae Yu, and Jinyoung Yeo. 2024. [Language models as compilers: Simulating pseudocode execution improves algorithmic reasoning in language](#)

- models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22471–22502, Miami, Florida, USA. Association for Computational Linguistics.
- Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. 2023. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *Transactions on Machine Learning Research*.
- Sara Court and Micha Elsner. 2024. Shortcomings of LLMs for low-resource translation: Retrieval and understanding are both the problem. In *Proceedings of the Ninth Conference on Machine Translation*, pages 1332–1354, Miami, Florida, USA. Association for Computational Linguistics.
- Antonios Dimakis, Stella Markantonatou, and Antonios Anastasopoulos. 2024. Dictionary-aided translation for handling multi-word expressions in low-resource languages. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2588–2595, Bangkok, Thailand. Association for Computational Linguistics.
- Matthew S. Dryer and Martin Haspelmath, editors. 2013. *WALS Online (v2020.4)*. Zenodo.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Micha Elsner and Jordan Needle. 2023. Translating a low-resource language using GPT-3 and a human-readable dictionary. In *Proceedings of the 20th SIGMORPHON workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 1–13, Toronto, Canada. Association for Computational Linguistics.
- Julen Etxaniz, Gorka Azkune, Aitor Soroa, Oier Lacalle, and Mikel Artetxe. 2024. Do multilingual language models think better in English? In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 550–564, Mexico City, Mexico. Association for Computational Linguistics.
- Mikel L. Forcada, Mireia Ginestí-Rosell, Jacob Nordfalk, Jimmy O’Regan, Sergio Ortiz Rojas, Juan Antonio Pérez-Ortiz, Felipe Sánchez-Martínez, Gema Ramírez-Sánchez, and Francis M. Tyers. 2011. Aperium: a free/open-source platform for rule-based machine translation. *Machine Translation*, 25:127–144.
- Gemini Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.
- Marjan Ghazvininejad, Hila Gonen, and Luke Zettlemoyer. 2023. Dictionary-based phrase-level prompting of large language models for machine translation. *Preprint*, arXiv:2302.07856.
- Michael Ginn, Mans Hulden, and Alexis Palmer. 2024a. Can we teach language models to gloss endangered languages? In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 5861–5876, Miami, Florida, USA. Association for Computational Linguistics.
- Michael Ginn, Lindia Tjautja, Taiqi He, Enora Rice, Graham Neubig, Alexis Palmer, and Lori Levin. 2024b. GlossLM: A massively multilingual corpus and pretrained model for interlinear glossed text. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 12267–12286, Miami, Florida, USA. Association for Computational Linguistics.
- Ping Guo, Yubing Ren, Yue Hu, Yunpeng Li, Jiarui Zhang, Xingsheng Zhang, and Heyan Huang. 2024. Teaching large language models to translate on low-resource languages with textbook prompting. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 15685–15697, Torino, Italia. ELRA and ICCL.
- Jonathan Hus and Antonios Anastasopoulos. 2024. Back to school: Translation using grammar books. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 20207–20219, Miami, Florida, USA. Association for Computational Linguistics.
- Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.
- Liguo Lan. 2016. *Annotated Zhuang Grammar Text*. Social Sciences Academic Press (China).
- Bryan Li, Tamer Alkhoul, Daniele Bonadiman, Nikolaos Pappas, and Saab Mansour. 2024. Eliciting better multilingual structured reasoning from LLMs through code. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5154–5169, Bangkok, Thailand. Association for Computational Linguistics.
- Emmy Liu, Graham Neubig, and Jacob Andreas. 2024. An incomplete loop: Instruction inference, instruction following, and in-context learning in language models. In *First Conference on Language Modeling*.
- Xiao Liu, Da Yin, Chen Zhang, Yansong Feng, and Dongyan Zhao. 2023. The magic of IF: Investigating causal reasoning abilities in large language models

- of code. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9009–9022, Toronto, Canada. Association for Computational Linguistics.
- Xiao Liu, Da Yin, Chen Zhang, Dongyan Zhao, and Yansong Feng. 2025. Eliciting and improving the causal reasoning abilities of large language models with conditional statements. *Computational Linguistics*, pages 1–38.
- Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. **Multilingual denoising pre-training for neural machine translation**. *Transactions of the Association for Computational Linguistics*, 8:726–742.
- Hongyuan Lu, Haoran Yang, Haoyang Huang, Dongdong Zhang, Wai Lam, and Furu Wei. 2024. **Chain-of-dictionary prompting elicits translation in large language models**. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 958–976, Miami, Florida, USA. Association for Computational Linguistics.
- Aman Madaan, Shuyan Zhou, Uri Alon, Yiming Yang, and Graham Neubig. 2022. **Language models of code are few-shot commonsense learners**. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1384–1403, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- OpenAI, Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. **Bleu: a method for automatic evaluation of machine translation**. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- Maja Popović. 2017. **chrF++: words helping character n-grams**. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.
- Matt Post. 2018. **A call for clarity in reporting BLEU scores**. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.
- Haritz Puerto, Martin Tutek, Somak Aditya, Xiaodan Zhu, and Iryna Gurevych. 2024. **Code prompting elicits conditional reasoning abilities in Text+Code LLMs**. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11234–11258, Miami, Florida, USA. Association for Computational Linguistics.
- Rita Ramos, Everlyn Asiko Chimoto, Maartje ter Hove, and Natalie Schluter. 2024. **Gramamt: Improving machine translation with grammar-informed in-context learning**. *arXiv preprint arXiv:2410.18702*.
- Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.
- Leipzig Glossing Rules. 2008. Conventions for interlinear morpheme-by-morpheme glosses. URL: <https://www.eva.mpg.de/lingua/resources/glossing-rules.php>.
- Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, Dipanjan Das, and Jason Wei. 2023. **Language models are multilingual chain-of-thought reasoners**. In *The Eleventh International Conference on Learning Representations*.
- Garrett Tanzer, Mirac Suzgun, Eline Visser, Dan Jurafsky, and Luke Melas-Kyriazi. 2024. **A benchmark for learning to translate a new language from one grammar book**. In *The Twelfth International Conference on Learning Representations*.
- Eline Visser. 2022. *A grammar of Kalamang*. Comprehensive Grammar Library. Language Science Press, Germany.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. **Chain-of-thought prompting elicits reasoning in large language models**. *Preprint, arXiv:2201.11903*.
- Jingyun Wei and Xiangzhou Qin. 2008. *Basic Course of Zhuang Language*. China Minzu University Press.
- Jingyun Wei and Xiaohang Qin. 2006. *General Introduction to Zhuang Language*. China Minzu University Press.
- An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.
- Zheng Xin Yong, Hailey Schoelkopf, Niklas Muenighoff, Alham Fikri Aji, David Ifeoluwa Adelani, Khalid Almubarak, M Saiful Bari, Lintang Sutawika, Jungo Kasai, Ahmed Baruwa, Genta Winata, Stella Biderman, Edward Raff, Dragomir Radev, and Vasilina Nikoulina. 2023. **BLOOM+1: Adding language support to BLOOM for zero-shot prompting**. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11682–11703, Toronto, Canada. Association for Computational Linguistics.
- Chen Zhang, Xiao Liu, Jiheng Lin, and Yansong Feng. 2024a. **Teaching large language models an**

unseen language on the fly. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 8783–8800, Bangkok, Thailand. Association for Computational Linguistics.

Kexun Zhang, Yee Choi, Zhenqiao Song, Taiqi He, William Yang Wang, and Lei Li. 2024b. [Hire a linguist!:](#) Learning endangered languages in LLMs with in-context linguistic descriptions. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 15654–15669, Bangkok, Thailand. Association for Computational Linguistics.

## A Introduction of Studied Languages

**The Zhuang Language** Zhuang is a group of Kra–Dai languages spoken primarily by the Zhuang people in Southern China, particularly in Guangxi Province and the neighboring regions of Yunnan and Guangdong. With over 16 million speakers, Zhuang is one of the largest minority languages in China. Zhuang is an isolating language, characterized by a lack of inflectional morphology. The current official writing system for Zhuang is the Latin script. In this study, we focus on Standard Zhuang, the officially standardized form of the language. Notably, current open-source and commercial language models show near-zero proficiency in processing Zhuang, highlighting its status as an underrepresented language that lacks support in existing LLMs.

**The Kalamang Language** Kalamang is an endangered language primarily used in the villages on the largest of the Karas islands off the west coast of the Bomberai Peninsula in Indonesian Papua. It belongs to the Greater West Bomberai language family. Kalamang has a small speaker population (less than 200 people) and was only recently documented (Visser, 2022; Tanzer et al., 2024). Written Kalamang uses the Latin script.

## B Dataset Collection and Statistics

### B.1 Grammar Book Extracts

In Figure 4, we provide a brief extract from the Zhuang grammar book (Wei and Qin, 2006, 2008), and in Figure 5 an extract from the Kalamang grammar book (Visser, 2022). Each figure begins with a textual grammar rule, followed by a set of parallel phrases or sentences as examples.

### B.2 Collection Details of Zhuang Data

**Rule Selection of ZHUANGRULES** From two grammar books, we extract suitable grammatical rules and their parallel sentence examples. A suitable rule should be concise and relatively independent, which means it does not rely on other rules as prerequisites. Afterward, we consult the dictionary at the end of the grammar book to build a lexicon for each Zhuang sentence in the dataset. Finally, we annotate fine-grained attributes for each rule, including action, difficulty, and domain, which will be discussed in the dataset statistics (Appendix B.4).

**IGT Generation for ZHUANGRULES** We use GPT-4o to generate an IGT for each Zhuang sentence in ZHUANGRULES. To obtain ICL examples, we collect 123 pieces of IGT from 《壮语语法标注文本》 (Annotated Zhuang Grammar Text; Lan, 2016). Each data point consists of a Zhuang sentence, its IGT, and its Chinese translation. The following is an example.

- (1) gou aeu aen laj  
1SG 要(want) CL-个(one) 下面(below)  
“我要下面那个 (I want the one below)”

We then use this collection of 123 IGT examples, together with a gloss list of 67 symbols, to guide GPT-4o in annotating each sentence in ZHUANGRULES.

We sample 20 IGTs generated by GPT-4o and check their quality. GPT-4o generates the correct symbols for 72% of the morphemes.

### Construction of Multi-Rule Testing Instances

We additionally collect a small set of data (96 test instances in total) from ZHUANGRULES that requires multiple rules for experiments. Specifically, we select suitable candidates from the parallel sentences in ZHUANGRULES and modify them to require two rules for translation. For instance, we modify sentences governed by a word order rule by inserting or replacing words related to a morphological rule. See Figure 6 for an example.

**Data Checking** We check the data in ZHUANGRULES to ensure it contains no information that names or uniquely identifies individuals, nor any offensive content.

### B.3 Collection Details of Kalamang Data

**Collection Process** The Kalamang grammar book in MTOB is written in a progressive style, where the understanding of most example sentences relies on multiple rules, which are difficult to annotate.

Given the infeasibility of engaging proficient Kalamang speakers in annotation, we leverage regular expressions to automatically match and extract certain contents from the book. Specifically, most parallel sentence examples in MTOB follow a three-line structure: the original Kalamang sentence, its IGT, and its English translation. We write regular expressions to capture this structure, using the paragraph preceding it as the corresponding grammar rule.

However, because the MTOB grammar book often requires multiple rules for each example sentence, and lacks annotations for specific rules, we exclude it from rule retrieval experiments. Additionally, the small number of parallel sentence examples (1.6 per rule on average) limits its utility in rule application experiments. Consequently, we focus primarily on grammar rules due to inadequate in-context learning sentence examples.

**Potential Issues** Note that the collected data might contain noise from two sources. First, many rules and their parallel examples in MTOB are not well-structured and cannot be captured by regular expressions. Second, rules in the Kalamang grammar book are often written in a progressive style (i.e., they directly reference parts of previously defined grammar rules without providing additional clarification, making it difficult to understand their meaning when viewed in isolation without giving those previously defined rules) and include extensive explanations for some parallel sentence examples, which complicates the rule descriptions.

These issues might hinder the performance of using collected grammar rules for translation, explaining why including grammar rules leads to a smaller improvement in Kalamang than in Zhuang.

#### B.4 Data Statistics

**Sizes and Lengths** ZHUANGRULES contains 109 rules in total, with an average length of 57.1 Chinese characters. These rules are paired with 608 Zhuang-Chinese parallel examples, of which 432 are words or phrases, and 176 are sentences. The average length of sentence examples is 8.4 Chinese characters or 5.4 Zhuang words. We also calculated the total token number for each element in ZHUANGRULES using the tokenizer for Qwen-2.5-72B-Instruct. The results are shown in Table 8.

From MTOB, we collect 97 rules for Kalamang, with an average length of 120.4 English words. These rules are paired with 152 Kalamang-English parallel examples. The average length of sentence examples is 9.2 English words or 10.9 Kalamang words.

**Additional Attributes** For each rule in ZHUANGRULES, we annotate several fine-grained attributes to gain deeper insights, including action, difficulty, and domain.

- **Action:** For each rule, we annotate the actions that are required when applying this rule for

Action		Domain	
Add	53	Morphology	28
Delete	6	Nominal Categories	16
Reorder	54	Nominal Syntax	1
Break	6	Verbal Categories	14
Select	22	Word Order	37
		Simple Clauses	7
		Complex Sentences	1
		Lexicon	6

Table 7: Number of rules for each type of action or domain in ZHUANGRULES.

Element	Total Tokens
Textual Rules	4386
Code Rules ( <i>Ver. For Retrieval</i> )	23,165
Code Rules	32,231
Lexicon	18819
Parallel Examples	6,992
IGTs	4,927
Text Rules + Lex. + 2 Ex.	13,973
Text Rules + Lex. + 2 Ex. + IGT	15,787
Code Rules + Lex. + 2 Ex.	41,818
Code Rules + Lex. + 2 Ex. + IGT	43,632

Table 8: Number of tokens for each elements in ZHUANGRULES.

Chinese-Zhuang translation. Possible actions include *add* (adding affixes to form a new word or adding some words to form a new phrase), *delete* (omitting certain words), *reorder* (reordering several words), *break* (breaking a word to several parts), and *select* (selecting a branch in the rule). As shown in Table 7, the two most common actions are *add* and *reorder*, which nearly half of the rules require.

- **Difficulty:** We categorize the rules into three levels: *easy*, *medium* and *hard*, based on the number of actions it involves and the degree of difference between Zhuang and Chinese reflected by this rule. Out of the 109 rules, there are 47 *easy*, 43 *medium*, and 19 *hard* rules. The average number of required operations is 1.2, 1.5, and 2.1 for the three levels of rules, respectively.
- **Domain:** Following WALS (Dryer and Haspelmath, 2013), a database of structural properties of languages, we label the rules with its linguistic domain, including *morphology*, *nominal categories*, *nominal syntax*, *verbal categories*, *word order*, *simple clauses*, *complex sentences*, and *lexicon*. As shown in Table 7, most of the rules deal with how the words are formed and ordered.

	za2zh		zh2za	
	BLEU	chrF++	BLEU	chrF++
No Rule	28.2	24.4	21.7	51.4
<i>Grammar Rules Only</i>				
Chinese Rule	51.4	50.5	51.8	78.7
English Rule	51.4	48.9	48.4	78.3
<i>Grammar Rules with Parallel Examples</i>				
Chinese Rule + 2 Ex.	<b>70.7</b>	<b>68.0</b>	80.7	91.0
English Rule + 2 Ex.	66.2	62.3	<b>81.6</b>	<b>92.3</b>

Table 9: MT performance of Qwen-2.5-72B-Instruct on Zhuang grammar rules written in different languages.

## C Implementation Details

**Conversion of Code Rules** Given our decomposition of using grammar books into two distinct steps—rule retrieval and rule application, we design different styles of code for the two steps, following the principle of using task-appropriate data. The code for rule retrieval shows the steps to examine whether the rule should be applied, while the code for rule application demonstrates the translation process. We provide an example of the code rules for rule retrieval in Table 11, and for rule application in Table 12.

### Prompts for Rule Retrieval and Application

For RULE-BY-RULE retrieval, we provide an example of prompt in Table 13. For rule application, we provide a prompt example in Table 14. We typically employ zero-shot learning without chain-of-thought (Wei et al., 2023) for efficiency considerations.

**Hyperparameters of Prompting** For the prompt-based method, we use greedy search without doing a hyperparameter search.

## D Additional Experimental Results

### D.1 Effect of Providing Varying Numbers of Rules

We present the complete results of MT performance by providing varying numbers of rules for Qwen-2.5-72B-Instruct. For Chinese-to-Zhuang translation, the BLEU score is shown in Figure 2. For Zhuang-to-Chinese translation, the BLEU score is shown in Figure 7. We observe similar phenomena on both translation directions.

### D.2 Effect of Grammar Rule Languages

As previous work shows that LLM performance varies when instructions are presented in different

	za2zh		zh2za	
	BLEU	chrF++	BLEU	chrF++
No Rule	31.2	28.3	22.1	49.1
Induced Rule	45.2	43.4	45.5	73.4
Gold Rule	51.4	50.5	51.8	78.7
Induced R. + 2 Ex.	67.9	64.2	73.0	85.5
Gold R. + 2 Ex.	<b>70.7</b>	<b>68.0</b>	<b>80.7</b>	<b>91.0</b>

Table 10: Translation performance of Qwen-2.5-72B-Instruct using induced and gold rules.

languages (Shi et al., 2023; Etxaniz et al., 2024), we examine how the language of rules affects translation. We translate the Chinese rules in ZHUANGRULES into English with GPT-4o and revise the translation manually.

We present the results of using the grammar rules written in English for translation in Table 9. The performance drop when using the English rules, compared to the original Chinese rules, is relatively small, indicating that the language of the grammar book has little impact on MT performance. We leave further analysis across languages in future studies.

### D.3 Induction of Descriptive Grammar Rules

Beyond being used for the study of translation, ZHUANGBENCH can be also used for the research of grammar induction.

**Experimental Setups** For each grammar rule in ZHUANGRULES, we ask LLMs to summarize a grammar rule based on its parallel sentence examples. We adopt two-shot ICL with Qwen-2.5-72B-Instruct,

Since directly evaluating the quality of the generated rules is challenging, we perform an extrinsic evaluation: we use the induced rules for machine translation and compare the results to those obtained using gold-standard rules.

**Results** As shown in Table 10, the grammar rules induced by LLMs are useful for translation. While the performance using induced rules is lower than using gold rules, it significantly outperforms the scenario without any rules, achieving improvements of 15.1% and 24.3% chrF++ for Zhuang-to-Chinese and Chinese-to-Zhuang translation, respectively. Additionally, pairing the induced rules with parallel examples further improves translation performance. We provide examples of the induced grammar rules in Table 16.

These results demonstrate the potential of using LLMs to generate grammar rules for XLR languages. Our pilot study focuses on clustered groups of simple parallel sentences sharing common grammatical features. Future work can explore the possibility of using LLMs to distill descriptive grammar from a larger, unordered corpus of parallel data.

## **E Case Study**

We provide examples of Zhuang-to-Chinese translation from different methods in Table 15. When no additional input is provided, the model fails to translate the sentence into Zhuang, instead generating sentences with irrelevant words that do not even appear in Zhuang dictionaries, highlighting the current limitations of LLMs in supporting the Zhuang language. However, when a lexicon is provided, the translation improves significantly, though it still suffers from hallucinations and syntax errors. When rules or parallel sentence examples are provided, the model can understand the rule or mimic the examples to perform the translation, yielding better results with both lexical and syntactical accuracy. Code rules further enhance the model's ability to understand and utilize grammar rules due to the similarity between code execution and grammar rule application, and the stepwise breakdown inherent in code rules helps LLMs better comprehend complex grammar rules, also leading to improved performance. Moreover, when the test instance requires multiple rules, including all the necessary rules enables the model to generate a better translation, while using only a subset of the required rules reduces performance.



---

```

def check_whether_apply(source_sentence, dictionary):
    """
    # 语法规则：在壮语中，单词“dwg”是系动词。在表示肯定判断的简单句中，“dwg”一般省略。然而，在表示否定时，“dwg”不可省略，且否定词“mbouj”需置于系动词“dwg”之前。(Grammar rule: In Zhuang, the word "dwg" is a copula. In affirmative simple sentences, "dwg" is usually omitted. However, in negative sentences, "dwg" cannot be omitted, and the negation word "mbouj" must be placed before the copula "dwg".)

    ## 检查是否需要应用此规则将汉语翻译为壮语的步骤如下：(The steps to check whether this rule needs to be applied in the translation are as follows:)
    1. 判断句子是否为判断句 (1. Determine if the sentence is a judgment sentence.)
    2. 若是判断句，返回True 表示需要应用该规则；否则，返回False (2. If the sentence is a judgment sentence, return True indicating the rule should be applied, otherwise return False.)
    """

    # 1. 判断句子是否为判断句 (1. Determine if the sentence is a judgment sentence.)
    if is_judgment_sentence(source_sentence):
        # 2. 若是判断句，返回True 表示需要应用该规则 (2. If the sentence is a judgment sentence, return True.)
        return True
    else: # 否则，返回False (Otherwise, return False)
        return False

```

---

Table 11: An example of the code rule for rule retrieval. The *text in italics* are the English translations. The code rules consist of two parts: (1) a concise comment outlining the steps for checking whether this rule needs to be applied, and (2) a pseudo-code function simulating the rule checking process.

---

```

def apply_rule(source_sentence, dictionary):
    """
    # 语法规则：在壮语中，单词“dwg”是系动词。在表示肯定判断的简单句中，“dwg”一般省略。然而，在表示否定时，“dwg”不可省略，且否定词“mbouj”需置于系动词“dwg”之前。(Grammar rule: In Zhuang, the word "dwg" is a copula. In affirmative simple sentences, "dwg" is usually omitted. However, in negative sentences, "dwg" cannot be omitted, and the negation word "mbouj" must be placed before the copula "dwg".)

    ## 应用此规则将汉语翻译为壮语的步骤如下：(The steps for applying this rule to translate Chinese into Zhuang are as follows:)
    1. 判断句子是否为否定句 (1. Determine if the sentence is negative.)
    2. 提取主语、谓语和宾语 (2. Extract the subject, predicate, and object.)
    3. 翻译主语、谓语和宾语 (3. Translate the subject, predicate, and object.)
    4. 根据句子类型（肯定或否定）调整系动词“dwg”和否定词“mbouj”的使用 (4. Adjust the use of copula "dwg" and negation word "mbouj" based on sentence type, affirmative or negative.)
    5. 拼接成完整的句子 (5. Combine to form the complete sentence.)
    """

    # 1. 判断句子是否为否定句 (1. Determine if the sentence is negative.)
    is_negative = "不是" in source_sentence
    # 2. 提取主语、谓语和宾语 (2. Extract the subject, predicate, and object.)
    subject, predicate, object_ = extract_subject_predicate_object(source_sentence)
    # 3. 翻译主语、谓语和宾语 (3. Translate the subject, predicate, and object.)
    subject_in_zhuang = translate(subject, dictionary)
    predicate_in_zhuang = translate(predicate, dictionary)
    object_in_zhuang = translate(object_, dictionary)
    # 4. 根据句子类型调整系动词和否定词 (4. Adjust the use of copula "dwg" and negation word "mbouj" based on sentence type, affirmative or negative.)
    if is_negative: # 否定句，使用“mbouj dwg” (Negative sentence, use "mbouj dwg")
        translated_sentence = f"subject_in_zhuang mbouj dwg predicate_in_zhuang object_in_zhuang"
    else: # 肯定句，省略“dwg” (Affirmative sentence, omit "dwg")
        translated_sentence = f"subject_in_zhuang predicate_in_zhuang object_in_zhuang"
    # 5. 返回完整的翻译句子 (5. Return the complete sentence.)
    return translated_sentence

```

---

Table 12: An example of the code rule for rule application. The *text in italics* are the English translations. The code rules consist of two parts: (1) a concise comment outlining the translation steps, and (2) a pseudo-code function simulating the translation process.

---

# 壮语是中国的一门少数民族语言。你是一名语言学家，以下是一条壮语语法规则的相关信息。给你一些需要翻译为壮语的汉语短语或句子，请根据该语法规则的内容，逐一检查翻译过程中是否需要使用该规则。你的回答中只应包含是否需要使用该规则的判断（“是”或“否”），不包含任何其他额外信息。*(Zhuang is a minority language in China. You are a linguist, and the following is a grammar rule for the Zhuang language. You are given some Chinese phrases or sentences to translate into Zhuang; please check whether this rule applies in the translation process. Your answer should only contain whether the rule needs to be applied ("yes" or "no"), and no other additional information.)*

## 语法规则: *(Grammar Rule:)*

### 语法规则描述: 在壮语中，单词“dwg”是系动词。在表示肯定判断的简单句中，“dwg”一般省略。然而，在表示否定时，“dwg”不可省略，且否定词“mbouj”需置于系动词“dwg”之前。*(Description of the Grammar Rule: In Zhuang, the word "dwg" is a copula. In affirmative simple sentences, "dwg" is usually omitted. However, in negative sentences, "dwg" cannot be omitted, and the negation word "mbouj" must be placed before the copula "dwg".)*

## 请检查下面的汉语短语或句子，判断是否需要使用该语法规则进行翻译：*(Please check the following Chinese phrase or sentence and determine whether the grammar rule needs to be applied in the translation:)*

### 汉语短语或句子: 他是我的父亲。*(Chinese phrase or sentence: He is my father.)*

### 在上面的短语或句子中，汉语词语“父亲”在壮语中的翻译是“daxboh”；汉语词语“他”在壮语中的翻译是“de”；汉语词语“我”在壮语中的翻译是“gou”。*(In the above phrase or sentence, the Chinese word "father" is translated into Zhuang as "daxboh"; "he" is translated as "de"; and "I" is translated as "gou".)*

---

Table 13: Prompt template for the RULE-BY-RULE retrieval method. The *italicized text* represents the English translations. When applying code rules, we directly replace the rule in the *Description of the Grammar Rule* with the code-format one.

---

# 壮语是中国的一门少数民族语言。你是一名语言学家，请根据给出的信息将汉语短语或句子翻译成壮语。你的回答应该只包含翻译结果，不要包含任何其他额外信息。*(Zhuang is a minority language in China. You are a linguist, please translate the given Chinese phrases or sentences into Zhuang. Your answer should only include the translation, without any additional information.)*

# 以下是一条关于壮语的语法规则：*(Below is a grammar rule for Zhuang:)*

## 语法规则: 在壮语中，单词“dwg”是系动词。在表示肯定判断的简单句中，“dwg”一般省略。然而，在表示否定时，“dwg”不可省略，且否定词“mbouj”需置于系动词“dwg”之前。*(Description of the Grammar Rule: In Zhuang, the word "dwg" is a copula. In affirmative simple sentences, "dwg" is usually omitted. However, in negative sentences, "dwg" cannot be omitted, and the negation word "mbouj" must be placed before the copula "dwg".)*

# 以下为该规则的一些例句及其IGT(Interlinear Glossed Text)，可以帮助你完成翻译：*(Here are some example sentences with their IGT (Interlinear Glossed Text), which can help you in the translation process:)*

## 例句1: *(Example 1:)*

字典为: {"de": "他", "daxboh": "父亲", "gou": "我"} *(Dictionary: {"de": "he", "daxboh": "father", "gou": "I"})*

壮语: De mbouj dwg daxboh gou.

IGT: 3sg NEG COP 父亲 1sg

汉语: 他不是我的父亲。*(Chinese: He is not my father.)*

## 例句2: *(Example 2:)*

字典为: {"daxmeh": "妈", "gou": "我", "vunz": "人", "laj mbanj": "乡下"} *(Dictionary: {"daxmeh": "mother", "gou": "I", "vunz": "person", "laj mbanj": "rural area"})*

壮语: Daxmeh gou vunz laj mbanj.

IGT: 母亲 1sg 人 乡下

汉语: 我妈是乡下人。*(Chinese: My mother is from the countryside.)*

## 请首先写出下面的壮语短语或句子的IGT，然后将其翻译成汉语：*(Please first write the IGT of the following Zhuang phrase or sentence, and then translate it into Chinese:)*

字典为: {"de": "他", "daxboh": "父亲", "gou": "我"} *(Dictionary: {"de": "he", "daxboh": "father", "gou": "I"})*

壮语: De daxboh gou.

## 所以，该壮语短语或句子的IGT和汉语翻译分别是：*(So, the IGT and Chinese translation for the Zhuang phrase or sentence are:)*

---

Table 14: Prompt template for grammar rule application with 2 parallel examples and IGTs. The *italicized text* represents the English translations. When applying code rules, we directly replace the rule in the *Description of the Grammar Rule* with the code-format one.

## 2. naengz “每” (Every)

ngoenznaengz 意为“每天”，这是一个由ngoenz (“天”)和naengz 组成的合成词。naengz 本身无实际意义，但附加在一些时间名词或动量词之后，便含有“每”的意思。例如: (*The word "ngoenznaengz" means "every day." It is a compound word formed by "ngoenz" ("day") and "naengz." The word "naengz" itself has no specific meaning, but when attached to certain time nouns or dynamic verbs, it conveys the meaning of "every". For example:*)

binaengz 每年 (every year)  
ndwennaengz 每月 (every month)  
haemhnaengz 每晚 (every night)  
baeznaengz 每次 (every time)

Figure 4: A simplified extract from the Zhuang grammar book. The text in italics are the English translations.

### 10.2.1 Proximal wa ‘PROX’

#### 10.2.1.1 Spatial use

Proximal demonstrative *wa* is prototypically used adnominally, pronominally and identificationally to indicate referents that are close to the speaker. (9) illustrates an adnominal and (10) a pronominal use of the proximal demonstrative, both in object position (hence the object form *wat*). The demonstrative in (9) refers to a woman in a picture in front of the speaker, with the speaker pointing at her. The demonstrative in (10) stands in for a fishing net the speaker is holding.

- (9) *ma enem wat=a tu*  
3SG woman PROX.OBJ=FOC hit  
'He hits this woman.' [stim6\_11:45]
- (10) *ki wat napaki=kin ye ge*  
2PL PROX.OBJ use=VOL or not  
'Are you going to use this or not?' [conv3\_1:59]

Figure 5: A simplified extract from the Kalamang grammar book.

**Grammar Rule 1:** 壮语代词 *gag* “自己” 与第一/第二/第三人称代词结合时，放在第一/第二/第三代词前或后均可，但意义各不相同。放在前面的表示“仅、只”的意思。放在后面的表示“自己”的意思。(*The Zhuang pronoun "gag" meaning "oneself". It can be combined with first, second, or third-person pronouns, and it can be placed either before or after the pronoun, but the meaning differs in each case. When placed before the pronoun, it means "only" or "just". When placed after the pronoun, it means "oneself"*)

**Grammar Rule 2:** 壮语时间名词的表达具有一定规律性，通过加入特定后缀可以表示特定某天的时间。后缀 *-bonz*、*-lwenz*、*-neix*、*-cog*、*-rawz* 分别表示“前天”、“昨天”、“今天”、“明天”和“后天”的时间。(*The expression of time nouns in Zhuang follows certain patterns, where specific suffixes can indicate the time of a particular day. The suffixes -bonz, -lwenz, -neix, -cog, and -rawz represent "the day before yesterday", "yesterday", "today", "tomorrow" and "the day after tomorrow" respectively.*)

Utilize **Grammar Rule 1** Utilize **Grammar Rule 2**

**Zhuang:** De *gag* ninz haemhlwenz.

**Lexicon:** {"de": "他", "haemh": "晚上", "ninz": "睡"}  
({"de": "he", "haemh": "at night", "ninz": "sleep"})

**Chinese:** 他昨天晚上自己睡

(*He sleeps by himself at yesterday night.*)

Figure 6: An testing instance that requires multiple grammar rules. To construct this, we modify sentences governed by Rule 1 (a rule about word order) by inserting words related to Rule 2 (a morphological rule). The text in italics are the English translations.

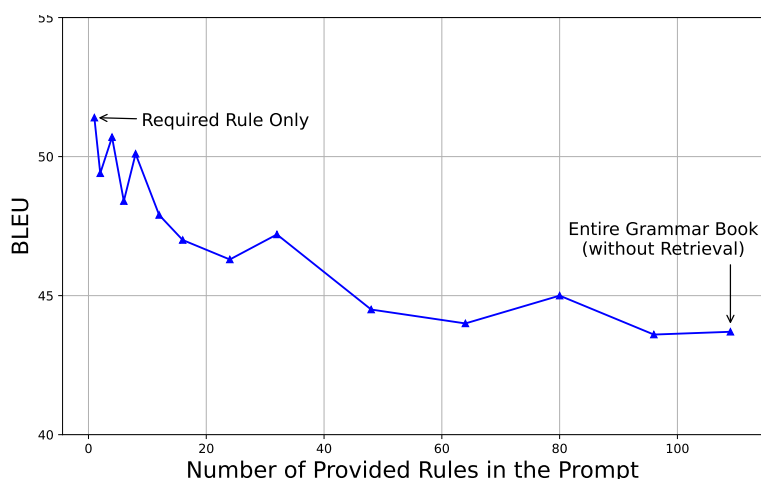


Figure 7: Performance of BLEU by rule number (Qwen-2.5-72B-Instruct, Zhuang → Chinese)

<i>Case 1 (Single Rule)</i>	
Grammar Rule:	壮语中表达“先做某事”用“..... gonq”，壮语中也有汉语借词sien“先”，需要将两者结合表达一个共同的意思。 <i>(In Zhuang, the expression “to do something first” is formed with “..... gonq”; Zhuang also has the Chinese loanword sien meaning “first,” which should be combined with gonq to express the meaning.)</i>
Input:	让她先讲一讲。 <i>(Let her speak first.)</i>
Lexicon:	{“让”: “hawj”, “她”: “de”, “讲”: “gangj”} (“let”: “hawj”, “she”: “de”, “speak”: “gangj”)
Gold:	Hawj de sien gangj gonq.
No Rule (w/o Lexicon):	Gou baez boux goz daen laj.
No Rule (w/ Lexicon):	Hawj de gangj.
Parallel Examples:	Hawj de sien gangj yijz gangj gonq.
Textual Rule:	Hawj de sien gonq gangj.
Code Rule:	Hawj de sien gangj gonq.
Textual Rule + 2 Examples:	Hawj de sien gangj gonq.
Code Rule + 2 Examples:	Hawj de sien gangj gonq.
<i>Case 2 (Multiple Rules)</i>	
Grammar Rule:	(1). 修饰语作中心语的修饰或限制时, 壮语中心语在前, 修饰语于后。 <i>(When modifiers function as modifiers or restrictors of the head, the head appears first, followed by the modifier in Zhuang.)</i> (2). “yaep ndeu” 意为“一会儿, 一下”, 在口语中使用频率较高, 用来表示时间的短暂性, 需要放在句尾。 <i>(“yaep ndeu” means “a while” or “a moment,” commonly used in spoken language to indicate the brevity of time, and should appear at the end of the sentence.)</i>
Input:	我看一会儿这本书。 <i>(I'll read this book for a while.)</i>
Lexicon:	{“我”: “gou”, “看”: “yawj”, “这”: “neix”, “本”: “bonj”, “书”: “saw”} (“I”: “gou”, “read”: “yawj”, “this”: “neix”, “measure word of books”: “bonj”, “book”: “saw”)
Gold:	Gou yawj bonj saw neix yaep ndeu.
No Rule (w/o Lexicon):	Gou baeq boux gaj goek.
No Rule (w/ Lexicon):	Gou yawj neix bonj saw haeuq.
Textual Rule 1 Only:	Gou yawj saw neix bonj.
Textual Rule 2 Only:	Gou yawj neix bonj saw yaep ndeu.
Both Textual Rules:	Gou yawj neix bonj saw yaep ndeu.
Combined Code Rule:	Gou yawj saw neix bonj yaep ndeu.

Table 15: Two examples of Zhuang-to-Chinese translation from different methods. The *text in italics* are the English translations.

Case 1	
Gold Grammar Rule:	“yied ..... yied .....”表示随着时间的推移或事物的发展，数量或程度逐渐增加或减少，类似于汉语中的“越来越.....”。 ( <i>"yied ..... yied ....." indicates a gradual increase or decrease in quantity or degree over time or with development, similar to the Chinese "more and more..."</i> .)
Input Parallel Example 1:	汉语：我们越走就越靠近他家了。 ( <i>Chinese: The farther we walk, the closer we get to his house.</i> ) Zhuang: Raeuz yied byaij yied dep laeng de lo. 字典为：{"我们": "raeuz", "走": "byaij", "靠近": "dep", "家": "laeng", "他": "de"} (Dictionary: {"we": "raeuz", "walk": "byaij", "approach": "dep", "house": "laeng", "his": "de"})
Input Parallel Example 2:	汉语：夏天到了，白天越来越长了。 ( <i>Chinese: Summer has arrived, and the days become longer.</i> ) Zhuang: Seizhah daengz lo, doengxngoenz yied daeuj yied raez. 字典为：{"夏天": "seizhah", "到": "daengz", "了": "lo", "白天": "doengxngoenz", "来": "daeuj", "长": "raez"} (Dictionary: {"summer": "seizhah", "arrive": "daengz", "completed": "lo", "daytime": "doengxngoenz", "become": "daeuj", "long": "raez"})
Induced Grammar Rule:	在壮语中，当需要表达某个状态或动作的程度逐渐加深或变化时，可以在两个相关的动词或形容词之间使用“yied...yied...”结构，以强调这种渐进的变化过程。 ( <i>In Zhuang, when expressing a gradual increase or change in a state or action, the "yied...yied..." structure can be used between two related verbs or adjectives to emphasize this progressive change.</i> )
Case 2	
Gold Grammar Rule:	“dox”+一般动词，表示动作行为是由甲乙双方相互进行的。 ( <i>"dox" + general verbs indicate that the action is performed by both parties in interaction.</i> )
Input Parallel Example 1:	汉语：对骂 ( <i>Chinese: To curse at each other.</i> ) Zhuang: doxndaq 字典为：{"骂": "ndaq"} (Dictionary: {"curse": "ndaq"})
Input Parallel Example 2:	汉语：互助 ( <i>Chinese: Mutual assistance.</i> ) Zhuang: doxbang 字典为：{"帮": "bang"} (Dictionary: {"help": "bang"})
Induced Grammar Rule:	在壮语中，为了表达两个主体之间的互动或相互作用的行为，可以在相关动词前添加前缀“dox”，以构成表示互动或相互作用的复合动词。 ( <i>In Zhuang, to express interactive or reciprocal actions between two subjects, the prefix "dox" can be added to relevant verbs to form compound verbs indicating interaction.</i> )
Case 3	
Gold Grammar Rule:	动词后边一般能在后面带上时态助词gvaq“过”，表示经历时态。但壮汉语在结构上略有不同，壮语的gvaq置于述宾词组之后，汉语的“过”则紧贴着动词之后。 ( <i>Verbs in Zhuang can typically be followed by the tense marker "gvaq" to indicate the experiential past. However, Zhuang and Chinese differ in structure: Zhuang places "gvaq" after the verb-object phrase, while in Chinese, "过" follows immediately after the verb.</i> )
Input Parallel Example 1:	汉语：我看过这本书。 ( <i>Chinese: I have read this book.</i> ) Zhuang: Gou yawj bonj saw neix gvaq. 字典为：{"我": "gou", "看": "yawj", "本": "bonj", "书": "saw", "这": "neix"} (Dictionary: {"I": "gou", "read": "yawj", "measure word of book": "bonj", "book": "saw", "this": "neix"})
Input Parallel Example 2:	汉语：他到过北京。 ( <i>Chinese: He has been to Beijing.</i> ) Zhuang: De bae daengz Bwzgingh gvaq. 字典为：{"他": "de", "去": "bae", "到": "daengz", "北京": "Bwzgingh"} (Dictionary: {"he": "de", "go": "bae", "arrive": "daengz", "Beijing": "Bwzgingh"})
Induced Grammar Rule:	在壮语中，表示过去经历的助词“gvaq”通常置于句子的末尾，用于表达动作已经发生或完成。 ( <i>In Zhuang, the experiential marker "gvaq" is usually placed at the end of the sentence to indicate that the action has already occurred or been completed.</i> )

Table 16: Examples of grammar rule inducted by Qwen-2.5-72B-Instruct. The text in italics are the English translations.