How Well Do Large Language Models Understand Syntax? An Evaluation by Asking Natural Language Questions

Anonymous ACL submission

Abstract

While recent advancements in large language models (LLMs) bring us closer to achieving artificial general intelligence, the question persists: *Do LLMs truly understand language, or do they merely mimic comprehension through pattern recognition?* This study seeks to explore this question through the lens of syntax, a crucial component of sentence comprehension. Adopting a natural language question-answering (Q&A) scheme, we craft questions targeting nine syntactic knowledge points that are most closely related to sentence comprehension. Experiments conducted on 24 LLMs suggest that most have a **limited** grasp of syntactic knowledge, exhibiting notable discrepancies across different syntactic knowledge points. In particular, questions involving *prepositional phrase attachment* pose the **greatest challenge**, whereas those concerning *adjectival modifier* and *indirect object* are **relatively easier** for LLMs to handle. Furthermore, a case study on the training dynamics of the LLMs reveals that **the majority of syntactic knowledge is learned during the initial stages of training**, hinting that simply increasing the number of training tokens may not be the '*silver bullet*' for improving the comprehension ability of LLMs.

1 Introduction

The rapid advancement of large language models (LLMs) has showcased their impressive abilities. Given a few exemplars or a set of instructions, LLMs can effectively handle a wide range of tasks, from traditional tasks like machine translation and summarization to more sophisticated, human-like activities such as solving mathematical problems, logical reasoning, and even planning. Distinctly different from their predecessors, which often required fine-tuning for specific tasks, LLMs are viewed as a significant stride towards artificial general intelligence (AGI).

`Sentence`: Pierre Vinken ***will join*** the board as a nonexecutive director Nov. 29.

`Question`: In the above sentence, the ***grammatical subject*** of "**will join**" is ____________.

`Options`:

- **A**. The board
- **B**. Pierre Vinken
- **C**. 61 years old
- **D**. A nonexecutive director

`Answer`: **B**

Figure 1: In this work, we aim to evaluate the syntactic understanding of LLMs by asking them questions phrased in natural language. This figure shows an example of syntactic knowledge questions presented in the natural language format that we used in this study.

Yet, even as we are surprised by the prowess of LLMs, questions about their true understanding of language arise. As black-boxes, do these models truly comprehend human language, or do they complete tasks by memorizing surface-level lexical patterns? Do LLMs understand sentences based on syntactic rules, or do they treat language as merely *a bag of words*?

Finding answers to these questions is of great importance to the LLM research community. Consider human-centric evaluation benchmarks, such as MMLU (Hendrycks et al., 2021) and AGIEval (Zhong et al., 2023), which comprise questions intended for humans, presuming test-takers' competent language understanding, an assumption that may not hold true for LLMs. Consequently, when an LLM errs in its response, discerning the root cause becomes convoluted. The error could be a manifestation of the model's knowledge gaps, an inability to reason, or simply a failure to understand the question due to a lack of syntactic knowledge. Measuring LLMs' syntactic knowledge is thus critical to understanding the true capabilities of LLMs.

To measure syntactic knowledge in LLMs, we must first determine **on which aspects of syntax we should focus**. In contrast to prior work that focuses on aspects such as forming grammatically correct sentences, explaining specific syntactic phenomena (Warstadt et al., 2019, 2020; Gauthier et al., 2020, *inter alia*), or depicting the hierarchical structure of sentences (Maudslay et al., 2020; Newman et al., 2021; Kim et al., 2023, *inter alia*), we concentrate on the comprehension aspect of syntax. Therefore, our study emphasizes the syntactic knowledge of grammatical relations, which are more closely related to sentence understanding. We evaluate the ability of LLMs to identify subjects, objects, complements and other syntactic roles in a sentence. Additionally, we also explore the ability of LLMs in resolving syntactic ambiguity. In total, we select nine syntactic knowledge to evaluate.

Then we turn to the methodology: **How should we evaluate syntactic knowledge in LLMs?** Prior work has proposed two main approaches: probing and prompting. However, these existed approaches have their limitations. The probing approach requires access to hidden states, which are not available for API-only models like the ChatGPT series, whereas the conventional prompting approach requires designing complex prompts and sophisticated decoding methods (Roy et al., 2022). In response to these limitations, we utilize a specific form of prompting, the natural language question-answering (Q&A) paradigm. This approach is a recently-mainstream and LLM-friendly evaluation method (Cobbe et al., 2021; Hendrycks et al., 2021; Zhong et al., 2023; Huang et al., 2023). For a thorough investigation, we design three question formats: True/False, Multiple Choice, and Fill in the Blank. An example is depicted in Figure 1.

We conducted extensive experiments on 24 LLMs from 6 distinct families, including the state-of-the-art `GPT4`, the open-source `LLaMA 1/2`, and other popular models, under both zero-shot and few-shot settings. Our findings indicate that while most LLMs have a partial grip on syntactic knowledge, `GPT4` demonstrates exceptional superiority in all tested scenarios. Closer examination showed that the prepositional phrase attachment (`PPA`) questions pose the greatest challenge, whereas adjectival modifier (`ADJ`) and indirect objects (`IO`) are comparatively simpler for LLMs to process. Interestingly, we also observe that alignment procedure exhibits potential benefits for `PPA` questions.

Additionally, a case study on `Baichuan2` explores how syntactic knowledge evolves throughout training. Our observations indicate that the majority of syntactic learning takes place in the early stages of training, suggesting that merely increasing the training tokens may not be the best way to improve syntactic knowledge.

In summary, our main contributions are as follows:

- We introduce a syntactic evaluation framework that evaluates LLMs' syntactic knowledge by asking LLMs natural language questions.
- Our comprehensive experiments across 24 LLMs reveal that most of LLMs are partially grasping syntactic knowledges.
- We dip into the learning curve of syntactic knowledge and find that the majority of this knowledge is acquired during the initial stages.

We hope that our research is a step towards a more comprehensive understanding of LLMs' strengths and limitations. Our code and dataset will be publicly available at `https://github.com`.

2 Design & Construction of Evaluation

In this study, we aim to investigate whether a LLM has essential syntax to understand a sentence. To this end, we introduce a novel syntactic evaluation framework, in which we *evaluate* LLMs by **asking them natural language questions**.

This section details the rationale behind our approach, outlines the core principles guiding our evaluation design, describes the process of crafting the questions, and discusses the methodology adopted in constructing the evaluation framework.

2.1 Motivation

The primary objective of this evaluation is to find a way to investigate whether a language model has essential syntax to understand a sentence.

The syntax of a language is the consensus of how to arrange words to express specific meanings. Only when words are arranged correctly can a sentence convey the writer's intended meaning. Similarly, only when the reader understands the syntax can they fully grasp the sentence's meaning. Therefore, the ability to understand a sentence is based on the syntactic knowledge of the reader.

2.2 Design Principles

Relevance to understanding The first principle is that the syntactic knowledge we investigate in our evaluation should be directly related to the understanding of a sentence. If a language model fails

Syntactic Knowledge Points	Abbr.	Example	#TF	#MC	#FITB
Grammatical **S**ubject	GS	**Desks** *are cleared* by John.	130	105	105
Subject **C**omplement	SC	John *is* **a teacher**.	130	85	85
Direct **O**bject	DO	John *gave* me **a book**.	150	145	145
Indirect **O**bject	IO	John *gave* **me** a book.	30	20	20
Main **V**erb **P**hrase	MVP	John **gave** me a book.	440‡	170	170
ADJectival modifier†	ADJ	I enjoy *the book* **John gave me**.	185	165	135
ADVerbial modifier (Adjunct)	ADV	I *read* the book **quickly**.	165	125	115
COordination	CO	We **will play** football and **watch** TV.	165	160	155
Prepositional **P**hrase **A**ttachment	PPA	I like **the book** *on my shelf*. I **hide** the book *on my shelf*.	110	100	100

Table 1: Syntactic knowledge points and the number of questions in our evaluation. †: We only consider post-modifier, such as relative clause and reduced relative clause in this work. ‡: The questions of main verb phrase in True/False are the same as those in surface subject, subject complement, direct object, and indirect object, so we directly reuse the questions of these four syntactic knowledge points and do not count them in the total number.

to identify this knowledge, it will probably fail to understand the sentence correctly.

Ease of Evaluation Our second principle is about the simplicity of the evaluation process. The notion for syntactic knowledges must be universal and easily comprehensible, thus precluding the necessity for specialized, academic, or domain-specific linguistic expertise. Additionally, the evaluation methodology should avoid the need to access a model's hidden states, which is not available for API-only models like the ChatGPT series. Lastly, the evaluation should leverage the model's strength in generating natural language responses rather than demanding strict structural outputs, like bracketed or even CoNLL-formatted strings.

2.3 Selection of Syntactic Knowledge

According to the Lexical-Functional Grammar theory, the syntactic structure of a sentence can be divided into two parts: a constituent structure (c-structure) and a functional structure (f-structure). The c-structure provides a hierarchical framework, illustrating how individual components sequentially combine to form a complete sentence. This can be analogized to a LEGO instruction manual for constructing a sentence. For example, the noun phrase "*I*" and the verb phrase "*am Batman*" can combine to form a sentence "*I am Batman*". On the other hand, the f-structure is represented as a series of key-value pairs, detailing the functions of phrases and words, identifying, such as, which phrase serves as the subject and which as the object. For example, in the sentence "*What I want is a car*", the f-structure is `Subject`: "*What I want*", `Object`: "*a car*" and etc.

Recall that our objective is to investigate whether a language model can use syntactic knowledge to identify the elements of a sentence in order to understand it, rather than to generate a syntactically correct sentence, which has been extensively studied in previous work (Warstadt et al., 2019, 2020; Gauthier et al., 2020). Therefore, we mainly focus on the f-structure. That is, we want to know whether LLMs can identify the subject, object, and other syntactic elements of a sentence. Besides the f-structure, we also explore the capability of LLMs in resolving syntactic ambiguity, another crucial factor influencing sentence comprehension. To this end, we also investigate two c-structure related syntactic knowledge: the coordination structure and the prepositional phrase attachment.

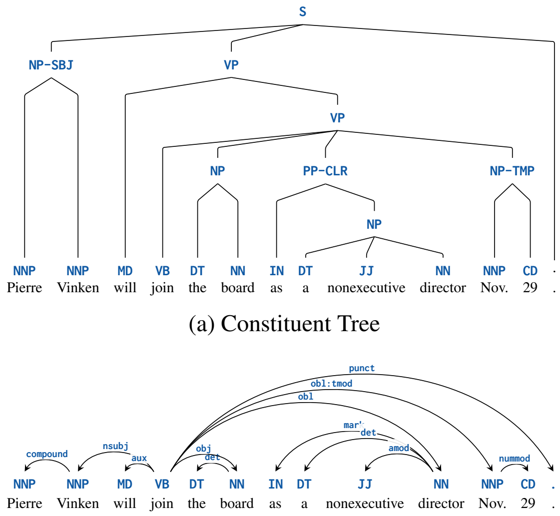
The full list of syntactic knowledge we investigate is shown in Table 1.

2.4 Selection of Paradigm

In line with the second design principle, we follow the recent mainstream approach of LLMs evaluating work, such as GSM8k (Cobbe et al., 2021), MMLU (Hendrycks et al., 2021), and AGIEval (Zhong et al., 2023), using **a question-answering (Q&A) paradigm**. That is, we pose a natural language question to the model as a prompt, and the model is expected to answer the question in natural language as well. We include three question types: True / False, Multiple Choice, and Fill in the Blank, for a holistic evaluation.

2.5 Design of Questions

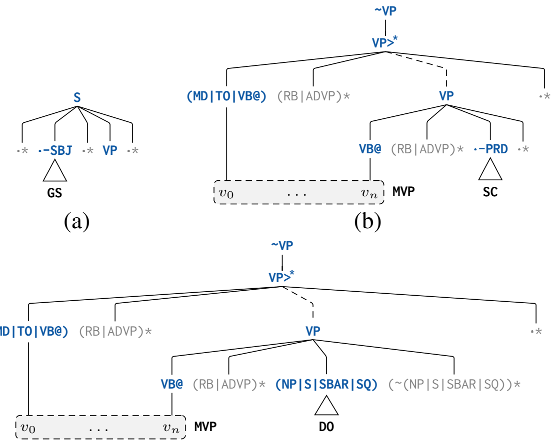
In the design of our questions, we adopted traditional syntactic concepts to guide our investigation into syntactic knowledge. The questions are



(a) Constituent Tree

(b) Dependency Tree

Figure 2: Two types of syntactic trees.



(a)

(b)

(c)

Figure 3: Three examples of syntactic patterns. "·" matches any pharse or word; "*" matches zero or more times horizontally; ">*" matches zero or more times recursively; "|" matches either the left or the right pattern; "~" is the negation of the pattern; "VB@" matches verb related part-of-speech tags, such as "VB", "VBZ".

structured such that the answers are phrases or full words from the sentence, mirroring the more natural human approach to responding to questions, rather than just the head word of the phrase. For example, for the sentence shown in Figure 2, when asked, "*What is the prepositional object of 'as'?*", most individuals are tended to answer with the complete phrase "*a nonexecutive director*," as opposed to the singular head word "*director*."

2.6 Construction

Instead of manually creating questions and answers, we propose to take advantage of existing syntactic annotations to automatically generate questions and answers. In this subsection, we briefly introduce the process of automatic syntactic information extraction and question generation.

Extracting Syntactic Information In this work, we extract syntactic information from the Penn Treebank (PTB) (Marcus et al., 1994), which is a widely used constituency treebank. An example of the constituency tree is shown in Figure 2a.

Why do we use constituency trees instead of dependency trees? Extracting syntactic information from a sentence based on its dependency tree, as shown in Figure 2b, where the relationship between words is explicitly annotated, might seem more straightforward. However, two main reasons prevent us from directly utilizing the dependency tree. Firstly, most existing dependency treebanks are automatically converted from constituency treebanks. This conversion might introduce errors that we are unaware of. Secondly, the dependency tree models the relationships between word pairs, making the extraction of answer phrases or full words difficult.

To extract syntactic information, we first learn the PTB guidelines carefully, figure out how syntactic information is annotated, and design patterns for each type of syntactic knowledge. Some of the patterns we design are shown in Figure 3.

Then, by searching for the patterns in a constituency tree, we extract the syntactic information of the corresponding sentence. For example, the pattern shown in Figure 3a matches the "S" node that has both an immediate child labeled with "-SBJ" function tag and an immediate "VP" child. We can then extract the immediate child with the "-SBJ" as the subject of the sentence "S".

Question Generation We manually design question templates for each type of questions and every syntactic knowledge point. Then, we use the extracted syntactic information to fill in the templates to generate questions. Along with the question, we also generate the meta information, such as the syntactic category (e.g., noun phrase, *that*-clause, etc.) of the answer and the words that fill in the placeholder, for the convenience of future use.

3 Experiments

3.1 Experimental Setup

Our experiments are conducted under two distinct settings: **Zero-shot** and **Few-shot**. In both settings,

	Zero-shot					Few-shot				
	TF	MC	FITB		OA	TF	MC	FITB		OA
	Acc.	Acc.	Acc.	F_1		Acc.	Acc.	Acc.	F_1	
`Random`	50.00	25.00	0.68	23.21	28.66	50.00	25.00	0.68	23.21	28.66
`Mistral 7B`	51.08	50.42	40.19	57.01	50.03	56.50	56.59	55.60	69.58	58.56
`Mistral 7B (Instruct)`	57.65	52.93	36.12	53.17	51.74	56.06	54.60	46.05	62.68	55.01
`Baichuan2 13B`	52.11	54.98	36.21	53.84	50.71	52.05	57.67	52.59	66.39	56.40
`Baichuan2 13B (Chat)`	59.53	55.91	26.60	46.05	50.59	57.12	57.46	44.69	60.83	55.78
`Falcon 40B`	52.68	48.56	27.57	45.11	45.86	57.65	54.23	46.34	62.07	55.36
`Falcon 40B (Instruct)`	58.03	48.37	29.22	45.65	47.95	55.77	53.71	46.22	62.39	54.59
`Llama 65B`	58.59	56.00	45.63	62.62	56.24	52.24	55.23	61.10	74.11	58.36
`Llama2 70B`	57.09	66.14	46.21	63.57	59.37	57.34	66.95	61.59	75.11	64.21
`Llama2 70B (Chat)`	57.00	61.58	42.33	60.30	56.63	60.09	68.65	55.86	70.63	64.00
`GPT3.5`	59.53	58.70	55.34	71.44	60.54	63.38	73.58	57.28	72.36	67.26
`GPT4`	**81.88**	**88.19**	**63.98**	**77.78**	**80.32**	**88.83**	**92.28**	**69.32**	**83.10**	**85.77**

Table 2: Main results of our evaluation.

the models are prompted to provide direct answers (referred to as the answer-only approach), **without** leveraging the Chain of Thought (CoT) technique[1]. The prompt in this work consists of there parts: 1) a brief introduction of the question type, 2) several exemplars if the model is under few-shot setting, and 3) the question itself. When answering the question, we first pose the sentence of which the question is asked, and then append the question to the sentence. Several prompt template examples for asking questions we use in this work are shown in Figure 6 in Appendix E.1.

3.2 Question Sampling

After question creation, we collected 3,538,818 questions. We observe that the number of questions for each syntactic knowledge point is extremely unbalanced[2]. The number of questions for the knowledge point of `MVP` is 248 times that of the knowledge point of `IO`. Therefore, we conduct a balanced down-sampling to ensure that each syntactic knowledge point has a similar number of questions.

Specifically, we first combine the question type, the syntactic knowledge point, and the syntactic category of the answer into a tuple. For each tuple, we randomly sample $k = 5$ questions from those associated with it to form the evaluation set. At the conclusion of this process, our test set comprises 3,170 questions, with detailed statistics presented in Table 1. Employing a similar approach but with a reduced sample size, we derived an exemplar set containing 1,300 questions.

[1]Due to the space limitations, we provide a detailed discussion of the CoT technique in Appendix E.2.

[2]The statistics of the questions are shown in Appendix B.

3.3 Evaluation Metrics

For True/False and Multiple Choice questions, we employ standard accuracy, adhering to conventions set by previous work. For Fill in the Blank questions, we utilize Accuracy (Acc.) and F_1 score (Question-wise averaging) as evaluation metrics. Finally, we report the overall performance (OA) as the average of the three question types:

$$\textbf{OA} = \frac{1}{3}\left(\textbf{TF}_{Acc.} + \textbf{MC}_{Acc.} + \frac{1}{2}(\textbf{FITB}_{Acc.} + \textbf{FITB}_{F_1})\right) \tag{1}$$

Due to the space limitations, we provide a detailed discussion of the metrics in Appendix C.1.

3.4 Selection of Large Language Models

We conduct comprehensive experiments on 24 large language models from 6 different families.

The 6 families are as follows: 1) **Mistral**, 2) **Baichuan2**, 3) **Falcon**, 4) **LLaMA**, 5) **LLaMA2**, and 6) **ChatGPT** series. More details can be found in Appendix D.

4 Results and Findings

In this section, we first present the experimental results for LLMs and provide a series of findings based on the results. We then conduct a case study on `Baichuan2` to further investigate the relationship between the number of training tokens and the model's performance. The detailed results of all models are presented in Appendix F.

4.1 Main Results

The main results are shown in Table 2 and the overall accuracy (OA) across different knowledge points is presented in Figure 3. From the results, we can observe several interesting findings:

Models	GS	SC	DO	IO	MVP	ADJ	ADV	PPA	CO
`Mistral 7B`	62.81	57.68	63.22	68.76	59.66	64.06	55.13	38.26	60.74
`Mistral 7B (Instruct)`	61.89	52.80	60.76	55.42	53.42	58.51	58.19	33.16	56.21
`Baichuan2 13B`	59.61	58.66	61.41	67.90	60.68	62.15	54.39	30.45	55.96
`Baichuan2 13B (Chat)`	63.81	58.52	59.97	65.04	57.31	61.78	50.79	33.13	56.05
`Falcon 40B`	61.38	55.45	57.21	64.90	60.36	60.11	50.36	36.05	55.71
`Falcon 40B (Instruct)`	57.50	56.17	57.40	58.26	60.78	62.55	49.54	36.55	51.67
`Llama 65B`	63.42	60.58	62.67	71.35	59.34	65.38	56.15	39.86	55.27
`Llama2 70B`	70.83	65.67	63.36	82.20	65.59	74.58	61.82	44.54	60.68
`Llama2 70B (Chat)`	68.86	56.22	67.14	68.76	70.36	75.04	60.00	49.72	56.33
`GPT3.5`	75.95	69.93	70.55	80.42	69.94	70.57	62.98	58.94	58.71
`GPT4`	**89.74**	**86.70**	**86.99**	**96.67**	**85.29**	**92.44**	**73.55**	**81.50**	**87.63**
Avg.	55.44	53.58	55.23	58.35	54.00	58.53	49.65	36.43	51.62

Table 3: Overall performance of each model under Few-shot setting at the knowledge point level.

I) LLMs is partially grasping syntax: As shown in Table 2 and 6 in Appendix F, the overall accuracy (OA) of all models larger than 1B is significantly higher than the random baseline, which indicates that LLMs do have the basic ability to understand syntax. However, only two models, `GPT4` and `GPT3.5`, have an OA greater than 60 in both settings, and only two other models, `Llama2 70B` and `Llama2 70B (Chat)`, have an OA higher than 60 on few-shot setting. This indicates that most LLMs can not answer the syntactic knowledge questions very well, and there is still a long way to go.

II) Few-shot beats Zero-shot in most cases: The zero-shot setting requires the model to understand the meaning of syntactic terms, such as "*subject*" and "*object*", and to identify the corresponding syntactic elements in the sentence. It is more difficult than the few-shot setting. As expected, compared to the few-shot setting, the zero-shot setting has a lower OA (from -2.88 to -11.42) on all models. The performance decline in Fill in the Blank questions is greater than that in True/False and Multiple Choice questions. It is worth noting that, there is one exception where some Chat/Instruct models have a higher accuracy in True/False questions on zero-shot setting than few-shot setting.

III) GPT4 shows superior performance: All results consistently show that `GPT4` outperforms other models by a large margin with an OA difference of 20.06 on zero-shot setting and 18.65 on few-shot setting. Even its results on the zero-shot setting are better than those of all other models in the few-shot setting. When we look at the results of different knowledge points, we can find that `GPT4` exceeds 85 OA on 7 out of 9 knowledge points on few-shot setting, among which the OA of indirect object (`IO`) are even higher than 95. Despite the superiority of `GPT4`, there are still some other models that outperform `GPT4` on some knowledge points. For example, when answering fill in the blank questions, `Llama2 70B` outperforms `GPT4` on the knowledge point of adverbial modifier (`ADV`) on both zero-shot and few-shot settings and coordination (`CO`) on zero-shot setting.

IV) `PPA` tops difficulty, `ADJ` and `IO` rank as easiest: Table 3 offers a granular analysis of results across different syntactic knowledge points. From the average results across all models, we can observe that the knowledge point of prepositional phrase attachment (`PPA`) is the most difficult one, with an OA of 36.43, while that of adjectival modifier (`ADJ`) and indirect object (`IO`) are the easiest ones, with an OA of 58.53 and 58.35, respectively.

V) Alignment procedure benefits `PPA` questions: From Table 3, we can observe that the most of Chat/Instruct models have a higher OA on `PPA` than their corresponding foundation models. For example, the OA of `Llama2 70B (Chat)` on `PPA` is 5.18 higher than that of `Llama2 70B`, while inferior on almost all other knowledge points. The same phenomenon also appears on `Baichuan2 13B` and `Baichuan2 13B (Chat)`. We suggest that this is because that the correct understanding of `PPA` is crucial for the chat task.

4.2 Training Dynamics for Knowledge Points: A Case Study on `Baichuan2`

Understanding *when* and *how* LLMs learn their knowledge is essential for developing LLMs (Müller-Eberstein et al., 2023). Therefore, we conduct a case study on `Baichuan2 7B` to explore the relationship between the pre-training process and the model's performance. `Baichuan2 7B` has been

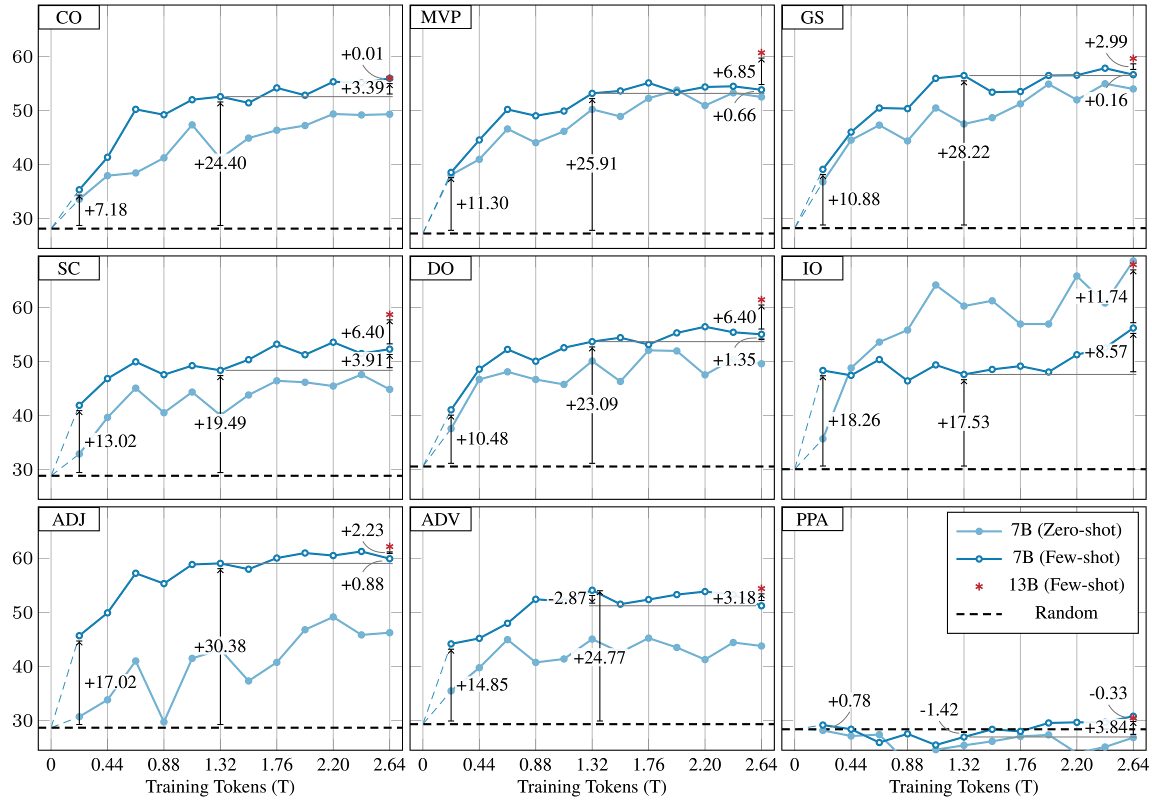


Figure 4: The overall scores of `BaiChuan2` intermediate checkpoints with different numbers of training tokens.

trained with a total of 2.64T tokens. Intermediate checkpoints were made publicly available after every 220B tokens trained.

As shown in Figure 4, the results reveal several trends common to most knowledge points: 1) There is a positive correlation between the number of training tokens and performance across most knowledge points: the more tokens trained, the better the performance. 2) After the initial training with 220B tokens, the model significantly exceeds the random baseline across most knowledge points, with improvements ranging from +7.18 to +18.26, except for `PPA`. 3) The most substantial performance gains occur during the first 1.32T tokens; beyond this point, the improvements are considerably smaller across most knowledge points (average improvement of 2.88 vs. 21.37 of the first 1.32T tokens).

However, there are interesting exceptions: 1) Performance on `PPA` remains low, which is close to the random baseline, across all three stages, indicating that merely increasing the number of training tokens does not nessarily improve performance on this knowledge point. Even when examining a larger model, `Baichuan2 13B`, we observe no significant performance gain on `PPA`. However, as mentioned in Finding V, alignment procedure has been shown to improve performance on this particular knowledge point. Therefore, how other model families effectively learn `PPA` and why human alignment is beneficial to solve `PPA` are intriguing topics for future research. 2) The zero-shot performance on the knowledge point of indirect objects (`IO`) is substantially higher than few-shot performance from the 440B tokens' training stage onward. A closer investigation reveals that the model is confused and misled by in-context exemplars, tending to answer based on previous exemplars that it mistakenly associates with direct objects, which is more common than indirect objects. This tendency **to overvalue in-context exemplars at the expense of the question itself** is a phenomenon also observed in other smaller models, such as `Falcon 1B/7B`, `Llama 7B/13B`, and `Llama2 7B/13B`, suggesting that smaller models may overly rely on in-context exemplars.

5 Related Work

5.1 Evaluation of Large Language Models

Recently, there has been a growing fascination with LLMs due to their remarkable performance across a wide spectrum of tasks. Evaluating these models serves a dual purpose by revealing both their capabilities and limitations. The results of the evaluation can offer valuable insights for refining and advancing LLMs. Typically, evaluations are designed to assess the ability to perform specific tasks. For example, GSM8k (Cobbe et al., 2021) evaluate the ability to perform mathematical reasoning, ToolLLM (Qin et al., 2023) evaluate the tool-use capabilities, and AGIEval (Zhong et al., 2023) use human-centric exams to evaluate the cognition and problem-solving abilities. Besides task-specific evaluations, numerous evaluation benchmarks (Hendrycks et al., 2021; Srivastava et al., 2022; Liang et al., 2022) have been proposed to assess generalization capabilities of LLMs. For example, HELM (Liang et al., 2022) evaluate prominent LLMs, covering a wide range of metrics, including model bias, efficiency, robustness, and more.

Our work belongs to the former category, specifically focusing on evaluating LLMs' linguistic comprehension capabilities.

5.2 Syntactic Knowledge in Language Models

Syntactic knowledge is a vast and complex topic, encompassing a wide range of aspects. These include forming grammatically correct sentences, explaining specific syntactic phenomena, and deciphering the meaning of sentences.

Many previous studies have focused on the first two aspects. They evaluate the syntactic knowledge of LLMs by constructing pairs of sentences, in which one is syntactically acceptable and the other is not. The model's task is to determine which sentence is grammatically correct. A representative work in this category is BLiMP (Warstadt et al., 2020), covering 67 syntactic phenomena, including subject-verb agreement and filler-gap dependencies.

In this work, we concentrate on the latter aspect: the ability to correctly interpret the structure and thereby understand the meaning of sentences. There are previous studies in this direction that propose various methods, broadly categorized into **probing** and **prompting methods**.

Probing methods are based on the premise that the syntactic knowledge required to understand a sentence should be reflected in the model's hidden states. These methods aim to uncover and extract the latent hierarchical structure from a model's hidden layers, believed to represent syntactic knowledge (Maudslay et al., 2020; Li et al., 2020; Newman et al., 2021; Zhao et al., 2023; Kim et al., 2023). A probe is essentially a function, such as a static similarity metric or a trainable neural network, that measures the syntactic distance between two tokens. If this distance is small, then the token pair is considered to have a syntactic relationship or belong to the same constituent. However, probing methods are limited to models with accessible hidden states, making API-based models unsuitable for probing.

Prompting methods are more flexible and applicable to any model supporting text generation. Most work in this category involves prompting the model to parse a sentence into a hierarchical structure containing the syntactic knowledge needed to understand the sentence (Roy et al., 2022; Bai et al., 2023; Lin et al., 2023). Designing effective prompts for complex syntactic tasks remains a challenge, often requiring constrained decoding methods to ensure the model's output is in the desired format (Roy et al., 2022). In contrast, our work employs a specific type of prompting: the natural language Q&A paradigm, a recently mainstream and LLM-friendly evaluation method (Cobbe et al., 2021; Hendrycks et al., 2021; Zhong et al., 2023; Huang et al., 2023). Thus, we bypass the need for designing complex prompts or decoding methods.

6 Conclusions

In this work, we propose **investigating the syntactic knowledge of LLMs by asking them natural language question answering**, aiming to answer the question of whether LLMs truly understand language or just mimic comprehension via pattern recognition and memorization. We crafted a series of questions focusing on nine syntactic knowledge points that are fundamental to sentence comprehension. Our experiments across 24 models suggest that LLMs have a ***basic*** ability to understand syntax, but their ability to correctly answer questions is ***limited***. Additionally, we find that the performance of LLMs varies greatly across different syntactic knowledge points, with prepositional phrase attachment being the ***most difficult*** and adjectival modifier and indirect object the ***easiest***. Finally, we conduct a case study on `Baichuan2` to investigate the training dynamics of syntactic knowledge. We observe that **the majority of syntactic knowledge is learned during the <u>early stages</u> of training**. This observation suggests that simply increasing the training tokens may not be the '*silver bullet*' for improving the comprehension ability of LLMs.

Limitations

This study is subject to several limitations.

The primary limitation stems from the indirect nature of our methodology, which lacks direct access to the model's hidden states and attention mechanisms. As such, it lacks the capability to inspect the model's '*neurons*' to determine how syntactic knowledge is stored and represented. However, this limitation is not unique to our work and is shared by the majority of existing studies on LLMs evaluation.

Additionally, our investigation covers only a select set of nine syntactic knowledge points. The field of syntax is vast, and numerous other phenomena warrant further examination to gain a comprehensive understanding of LLMs' capabilities. Moreover, the scope of our syntactic evaluation is confined to the English language, meaning that the findings may not be generalizable across different languages, such as Chinese.

Lastly, our experimental setup was limited to models with fewer than 70 billion parameters due to resource constraints. Thus, the behaviors and performance of larger, potentially more capable models remain unexplored in our study.

Ethics Statement

We have diligently endeavored to ensure that our work adheres to high ethical standards.

Dataset: The dataset employed in this study is the Penn Treebank (LDC99T42), accessed under the LDC license. In compliance with this license, we are not permitted to redistribute the data. Therefore, for researchers who have access to the Penn Treebank, we provide only the code necessary to reconstruct the dataset utilized in our study for the purpose of reproducibility. Note that the questions generation process we used is fully automatic, and it will not increase any information that names or uniquely identifies individual people or offensive content.

Labor Considerations: All human labor involved in this study, which includes designing extraction patterns, formulating question templates, verifying extracted information, and reviewing generated questions, was performed voluntarily by the authors. This work was conducted with a commitment to ethical research practices, ensuring fairness and respect for all contributors.

Consequently, we believe that our work aligns with the ethical standards of the ACL community.

References

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. The falcon series of open language models. *ArXiv preprint*, abs/2311.16867.

Xuefeng Bai, Jialong Wu, Yulong Chen, Zhongqing Wang, and Yue Zhang. 2023. Constituency parsing using llms. *ArXiv preprint*, abs/2310.19462.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *ArXiv preprint*, abs/2110.14168.

Jon Gauthier, Jennifer Hu, Ethan Wilcox, Peng Qian, and Roger Levy. 2020. SyntaxGym: An online platform for targeted evaluation of language models. In *Proceedings of ACL*, pages 70–76, Online.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *Proceedings of ICLR*.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, Yao Fu, Maosong Sun, and Junxian He. 2023. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. *ArXiv preprint*, abs/2305.08322.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *ArXiv preprint*, abs/2310.06825.

Najoung Kim, Jatin Khilnani, Alex Warstadt, and Abdelrahim Qaddoumi. 2023. Reconstruction probing. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8240–8255, Toronto, Canada.

Huayang Li, Lemao Liu, Guoping Huang, and Shuming Shi. 2020. On the branching bias of syntax extracted from pre-trained language models. In *Proceedings of EMNLP*, pages 4473–4478, Online.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D. Manning, Christopher Ré, Diana Acosta-Navas, Drew A. Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue Wang, Keshav Santhanam, Laurel J. Orr, Lucia Zheng, Mert Yüksekgönül, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri S. Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. 2022. Holistic evaluation of language models. *ArXiv preprint*, abs/2211.09110.

Boda Lin, Xinyi Zhou, Binghao Tang, Xiaocheng Gong, and Si Li. 2023. Chatgpt is a potential zero-shot dependency parser. *ArXiv preprint*, abs/2310.16654.

Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. 1994. The Penn Treebank: Annotating predicate argument structure. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

Rowan Hall Maudslay, Josef Valvoda, Tiago Pimentel, Adina Williams, and Ryan Cotterell. 2020. A tale of a probe and a parser. In *Proceedings of ACL*, pages 7389–7395, Online.

Max Müller-Eberstein, Rob van der Goot, Barbara Plank, and Ivan Titov. 2023. Subspace chronicles: How linguistic information emerges, shifts and interacts during language model training. In *Proceedings of EMNLP*, pages 13190–13208, Singapore.

Benjamin Newman, Kai-Siang Ang, Julia Gong, and John Hewitt. 2021. Refining targeted syntactic evaluation of language models. In *Proceedings of NAACL-HLT*, pages 3710–3723, Online.

OpenAI. 2023. GPT-4 technical report. *ArXiv preprint*, abs/2303.08774.

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

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. 2023. Toolllm: Facilitating large language models to master 16000+ real-world apis. *ArXiv preprint*, abs/2307.16789.

Subhro Roy, Sam Thomson, Tongfei Chen, Richard Shin, Adam Pauls, Jason Eisner, and Benjamin Van Durme. 2022. Benchclamp: A benchmark for evaluating language models on semantic parsing. *ArXiv preprint*, abs/2206.10668.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, Alice Xiang, Alicia Parrish, Allen Nie, Aman Hussain, Amanda Askell, Amanda Dsouza, Ameet Rahane, Anantharaman S. Iyer, Anders Andreassen, Andrea Santilli, Andreas Stuhlmüller, Andrew M. Dai, Andrew La, Andrew K. Lampinen, Andy Zou, Angela Jiang, Angelica Chen, Anh Vuong, Animesh Gupta, Anna Gottardi, Antonio Norelli, Anu Venkatesh, Arash Gholamidavoodi, Arfa Tabassum, Arul Menezes, Arun Kirubarajan, Asher Mullokandov, Ashish Sabharwal, Austin Herrick, Avia Efrat, Aykut Erdem, Ayla Karakas, and et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *ArXiv preprint*, abs/2206.04615.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *ArXiv preprint*, abs/2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *ArXiv preprint*, abs/2307.09288.

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.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *TACL*, 7:625–641.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of EMNLP*, pages 38–45, Online.

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, Fan Yang, Fei Deng, Feng Wang, Feng Liu, Guangwei Ai, Guosheng Dong, Haizhou Zhao, Hang Xu, Haoze Sun, Hongda Zhang, Hui Liu, Jiaming Ji, Jian Xie, Juntao Dai, Kun Fang, Lei Su, Liang Song, Lifeng Liu, Liyun Ru, Luyao Ma, Mang Wang, Mickel Liu, MingAn Lin, Nuolan Nie, Peidong Guo, Ruiyang Sun, Tao Zhang, Tianpeng Li, Tianyu Li, Wei Cheng, Weipeng Chen, Xiangrong Zeng, Xiaochuan Wang, Xiaoxi Chen, Xin Men, Xin Yu, Xuehai Pan, Yanjun Shen, Yiding Wang, Yiyu Li, Youxin Jiang, Yuchen Gao, Yupeng Zhang, Zenan Zhou, and Zhiying Wu. 2023. Baichuan 2: Open large-scale language models. *ArXiv preprint*, abs/2309.10305.

Haoyu Zhao, Abhishek Panigrahi, Rong Ge, and Sanjeev Arora. 2023. Do transformers parse while predicting the masked word? *ArXiv preprint*, abs/2303.08117.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. 2023. Agieval: A human-centric benchmark for evaluating foundation models. *ArXiv preprint*, abs/2304.06364.

A Question Templates

Some of the question templates we designed (for the grammatical subject knowledge point) are shown in Figure 5.

B Original Question Distribution

The original distribution of the questions we built is shown in Table 4. From this table, we can see that the distribution of each syntactic knowledge point is imbalanced. The most common syntactic knowledge point is the main verb phrase, which accounts for 23.55% of all the questions, while the least common syntactic knowledge point is the indirect object, which only accounts for 0.09%.

C Evaluation

C.1 Evaluation Metrics

Notably, compared to prior studies, we adopt a stricter F_1 score, in which we require that words in the predicted answer align in the same order as those in the ground truth answer. To mitigate any potential issues arising from tokenization and punctuation discrepancies, we employ `NLTK`[3] (Bird et al., 2009) to re-tokenize then discard all punctuation before computing scores.

D Model Details

The information about the models we evaluated in this work is shown in Table 5.

Mistral series: Mistral (Jiang et al., 2023) is Mistral AI's first Large Language Model (LLM), a transformer model especially suited for NLP applications. It's trained on a vast dataset of text and code, enabling it to generate text, translate languages, produce creative content, and answer questions instructively. Mistral 7B, with 7.24 billion parameters, outperforms LLaMA 2 13B on all benchmarks and LLaMA 30B on many other benchmarks.

Baichuan2 series: The newest open-source and commercially available large language model series from Baichuan Inc. This series comprises four models: a 7B and a 13B foundation model, each with their corresponding chat versions (Yang et al., 2023). The `Baichuan2 7B` model is one of the few models that publicly release intermediate checkpoints, which facilitates our case study of the training dynamics of syntactic knowledge.

[3]https://www.nltk.org/

Falcon series: A series of large language models published by TII, trained on the Refined Web Dataset. This series includes three models with parameter sizes of 1B, 7B, and 40B. The 7B and 40B versions also have their corresponding instruction-tuned variants (Almazrouei et al., 2023).

LLaMA series: One of the most popular large language model series from Meta, which has been used in various works. This series includes four models with parameter sizes of 7B, 13B, 30B, and 65B (Touvron et al., 2023a).

LLaMA2 series: The new generation of the LLaMA series, trained on a cleaner and larger dataset. This series consists of three models with parameter sizes of 7B, 13B, and 70B, each with their corresponding chat versions (Touvron et al., 2023b).

ChatGPT series: Currently regarded as the most powerful large language model series, developed by OpenAI. However, most models in this series are accessible as pay-to-use, API-only models. For our experiments, we focused on two chat versions from this series: '`gpt-3.5-turbo-0613`' (Ouyang et al., 2022) and '`gpt-4-0613`' (OpenAI, 2023).

D.1 Implementation Details

For GPT series, we use the official Python API to access the models. We set the temperature to 0 and maximum length to 256 for "Fill in the Blank" questions and 10 for "True/False" and "Multiple Choice" questions. Other hyper-parameters are remained as default.

For other open-sourced models, we use the `transformers` library (Wolf et al., 2020) to access them. We **do not fine-tune** any of these models. If the model creator provides the special generation function, such as "`chat()`" in the `Baichuan2` series, we directly use it, otherwise we use the "`generate()`" function. The hyper-parameters are set to the same as the GPT series.

We use the same prompt for all the models, if the model creator does not provide a suggested prompt.

In few-shot experiments, for each question, we randomly select 5 exemplars having the same syntactic knowledge point and question type as the question has.

We run all the experiments with three random seeds, which will affect the exemplars selected for each question, and report the average results. The

True/False

In the above sentence, the grammatical subject of "`{verb_phrase}`" is "`{correct_answer}`".

`<NEG>` In the above sentence, the grammatical subject of "`{verb_phrase}`" is not "`{correct_answer}`".

In the above sentence, the grammatical subject of "`{verb_phrase}`" is "`{incorrect_answer}`".

`<NEG>` In the above sentence, the grammatical subject of "`{verb_phrase}`" is not "`{incorrect_answer}`".

Multiple Choice

In the above sentence, which of the following is the grammatical subject of "`{verb_phrase}`"?

```
<option_A>:={correct_answer}
<option_B>:={incorrect_answer_1}
<option_C>:={incorrect_answer_2}
<option_D>:={incorrect_answer_3}
[Randomly shuffle the options]
```

Fill in the Blank

In the above sentence, the grammatical subject of "`{verb_phrase}`" is ____________.

Figure 5: Question templates used for generating questions for the grammatical subject knowledge point. The `<NEG>` tag is used to indicate the negative form of the question, and will be *removed* when generating the question.

Syntactic Knowledge Points	Abbr.	#TF	#MC	#FITB	#total	Ratio (%)
Grammatical **S**ubject	GS	426,832	106,708	106,708	640,248	14.93
Subject **C**omplement	SC	59,984	14,996	14,996	89,976	2.10
Direct **O**bject	DO	261,320	65,330	65,330	391,980	9.14
Indirect **O**bject	IO	2,716	679	679	4,074	0.09
Main **V**erb **P**hrase	MVP	750,852[‡]	129,669	129,669	1,010,190	23.55
ADJectival modifier[†]	ADJ	587,968	67,865	58,401	714,234	16.65
ADVerbial modifier (Adjunct)	ADV	385,406	77,439	40,268	503,113	11.73
COordination	CO	319,492	33,405	19,594	372,491	8.68
Prepositional **P**hrase **A**ttachment	PPA	375,576	93,894	93,894	563,364	13.13

Table 4: Syntactic knowledge points in our evaluation. †: We only consider post-modifier, such as relative clause and reduced relative clause in this work. ‡: The questions of main verb phrase in True/False are the same as those in surface subject, subject complement, direct object, and indirect object, so we directly reuse the questions of these four syntactic knowledge points and do not count them in the total number of questions.

only exception is that we only run with one random seed on the pay-to-use GPT models, due to the high price of using them.

E Prompt Details

E.1 General Prompt

The general prompt for foundational models under zero-shot and few-shot settings is shown in Figure 6. For models like Falcon-Instruct and LLaMA2-Chat, which have their own special prompt format, we adjust the general prompt to fit their format accordingly.

E.2 The Problem of CoT

Our decision to exclude the Chain of Thought (CoT) (Wei et al., 2022) setting is grounded in two primary reasons. Firstly, in most instances, discerning the syntactic structure of a sentence does not require complex reasoning. Secondly, preliminary tests revealed that many models, particularly the less complex ones, struggled to generate coherent chains of thought tailored to our syntactic knowledge questions. Often, these models repetitively produce phrases like "The object of the `XXX` is `YYY`," extending up to the preset maximum generation length.

True/False (Prompt for Fill in the Blank questions is similar to this)

The following are true or false questions, please answer them with “True” or “False”.\n
Sentence: <sentence>\n
Question: <question>\n
Answer: The answer is “

Multiple Choice

The following are multiple choice questions, please answer them with “A”, “B”, “C”, or “D”.\n
Sentence: <sentence>\n
Question: <question>\n
Options: \n A. <option_A>\n B. <option_B>\n C. <option_C>\n D. <option_D>\n
Answer: The answer is “

(a) General Prompt for Foundational Models under Zero-shot Setting

True/False (Prompt for Fill in the Blank questions is similar to this)

The following are true or false questions (with answers):\n
Sentence: <exemplars[1].sentence>\n
Question: <exemplars[1].question>\n
Answer: The answer is “<exemplars[1].answer>”\n
...[exemplars omitted for brevity]...
Sentence: <exemplars[k].sentence>\n
Question: <exemplars[k].question>\n
Answer: The answer is “<exemplars[k].answer>”\n
Sentence: <sentence>\n
Question: <question>\n
Answer: The answer is “

Multiple Choice

The following are multiple choice questions (with answers):\n
Sentence: <exemplars[1].sentence>\n
Question: <exemplars[1].question>\n
Options: \n A. <exemplars[1].option_A>\n B. <exemplars[1].option_B>\n
C. <exemplars[1].option_C>\n D. <exemplars[1].option_D>\n
Answer: The answer is “<exemplars[1].answer>”\n
...[exemplars omitted for brevity]...
Sentence: <exemplars[k].sentence>\n
Question: <exemplars[k].question>\n
Options: \n A. <exemplars[k].option_A>\n B. <exemplars[k].option_B>\n
C. <exemplars[k].option_C>\n D. <exemplars[k].option_D>\n
Answer: The answer is “<exemplars[k].answer>”\n
Sentence: <sentence>\n
Question: <question>\n
Options: \n A. <option_A>\n B. <option_B>\n C. <option_C>\n D. <option_D>\n
Answer: The answer is “

(b) General Prompt for Foundational Models under Few-shot Setting

Figure 6: Prompt templates used in this work.

Model	Creator	#Parameters	Open-sourced
Mistral series			
Mistral-7B-v0.1	Mistral AI	7.24B	✓
Mistral-7B-Instruct-v0.1			
Baichuan2 series			
Baichuan2-7B-Base	Baichuan	7.51B	✓
Baichuan2-7B-Chat			
Baichuan2-13B-Base		13.90B	
Baichuan2-13B-Chat			
Falcon series			
falcon-rw-1b	TII	1.31B	✓
falcon-7b		6.92B	
falcon-7b-instruct			
falcon-40b		41.30B	
falcon-40b-instruct			
LLaMA series			
llama-7b	Meta	6.78B	✓
llama-13b		13.02B	
llama-30b		32.53B	
llama-65b		65.29B	
LLaMA2 series			
llama-2-7b	Meta	6.74B	✓
llama-2-7b-chat			
llama-2-13b		13.02B	
llama-2-13b-chat			
llama-2-70b		68.98B	
llama-2-70b-chat			
ChatGPT series			
gpt-3.5-turbo-0613	OpenAI	unknown	✗
gpt-4-0613			

Table 5: Models evaluated in this work. "#Parameters" is the number of parameters of the model. "Open-sourced" indicates whether the model is open sourced.

F Detailed Results

The results of all the models under all the settings are shown in Table 6. The correlation between the difficulty metrics is shown in Table 7. The overall accuracy of each model under each zero-shot and few-shot setting is shown in Table 8 and Table 9, respectively. The detailed performance of each model under each zero-shot and few-shot setting is shown in Table 10 and Table 12, respectively.

F.1 More Findings

Parameter size impacts performance differently: The relationship between parameter size and model performance is depicted in Figure 7. Within individual families, there's a general trend that aligns performance with parameter size: larger models tend to achieve better results. However, when comparing across different families, this correlation is not always consistent. For instance, the "Baichuan2 7B" model outperforms all 13B models in Multiple Choice questions.

Inconsistent knowledge generalize across question types: When we compare the metrics of different question types, we can find that the knowledge does not generalize well across different question types. First, we observe that when the model has a high performance on one question type, it does not mean that it will also have a high performance on other question types. For example, as shown in Table 2, even when Baichuan2 13B has outperformed random baseline by a large margin on Fill in the Blank questions, in which the model is required to generate the text of the answer, its OA on True/False questions is merely 2.05 higher than the random baseline. Second, we observe that the correlation between the performance on different question types is not consistent. The Kendall's τ and Pearson's r correlation coefficients are shown in Table 7 in Appendix F. The results indicate that correlations between the performance on True/False and other question types are all lower than 0.8, meaning that there is no strong correlation between the performance on True/False and

	Zero-shot					Few-shot				
	TF	MC	FITB		OA	TF	MC	FITB		OA
	Acc.	Acc.	Acc.	F_1		Acc.	Acc.	Acc.	F_1	
Random	50.11	24.03	0.42	22.20	28.48	49.42	24.62	0.68	23.21	28.66
Mistral 7B	51.08	50.42	40.19	57.01	50.03	56.50	56.59	55.60	69.58	58.56
Baichuan2 7B	53.99	47.35	31.65	48.28	47.10	50.61	52.81	46.86	62.34	52.67
Baichuan2 13B	52.11	54.98	36.21	53.84	50.71	52.05	57.67	52.59	66.39	56.40
Falcon 1B	51.46	24.00	5.05	16.34	28.72	49.64	25.76	17.77	36.27	34.14
Falcon 7B	50.52	25.77	16.60	33.03	33.70	47.07	27.53	26.63	43.67	36.59
Falcon 40B	52.68	48.56	27.57	45.11	45.86	57.65	54.23	46.34	62.07	55.36
Llama 7B	49.20	30.88	23.79	40.33	37.38	48.35	33.61	37.47	53.92	42.55
Llama 13B	49.39	41.67	24.85	39.93	41.15	48.86	36.53	45.01	60.90	46.12
Llama 30B	55.96	43.91	33.88	50.03	47.27	50.89	48.62	55.18	69.87	54.01
Llama 65B	58.59	56.00	45.63	62.62	56.24	52.24	55.23	61.10	74.11	58.36
Llama2 7B	53.52	34.14	23.01	38.37	39.45	48.73	35.19	42.72	58.08	44.77
Llama2 13B	53.62	41.86	29.81	44.39	44.19	54.46	41.92	51.65	66.40	51.80
Llama2 70B	57.09	66.14	46.21	63.57	59.37	57.34	66.95	61.59	75.11	64.21
Mistral 7B(Instruct)	57.65	52.93	36.12	53.17	51.74	56.06	54.60	46.05	62.68	55.01
Baichuan2 7B(Chat)	49.77	45.49	24.27	43.21	43.00	55.15	54.26	44.47	63.22	54.42
Baichuan2 13B(Chat)	59.53	55.91	26.60	46.05	50.59	57.12	57.46	44.69	60.83	55.78
Falcon 7B(Instruct)	51.55	27.63	11.94	23.71	32.34	53.65	28.59	19.19	36.01	36.61
Falcon 40B(Instruct)	58.03	48.37	29.22	45.65	47.95	55.77	53.71	46.22	62.39	54.59
Llama2 7B(Chat)	54.74	45.40	21.36	38.72	43.39	52.14	48.68	29.64	47.78	46.51
Llama2 13B(Chat)	57.00	47.91	26.12	45.42	46.89	51.83	51.47	44.47	62.22	52.21
Llama2 70B(Chat)	57.00	61.58	42.33	60.30	56.63	60.09	68.65	55.86	70.63	64.00
GPT3.5	59.53	58.70	55.34	71.44	60.54	63.38	73.58	57.28	72.36	67.26
GPT4	**81.88**	**88.19**	**63.98**	**77.78**	**80.32**	**88.83**	**92.28**	**69.32**	**83.10**	**85.77**

Table 6: Main results of our evaluation.

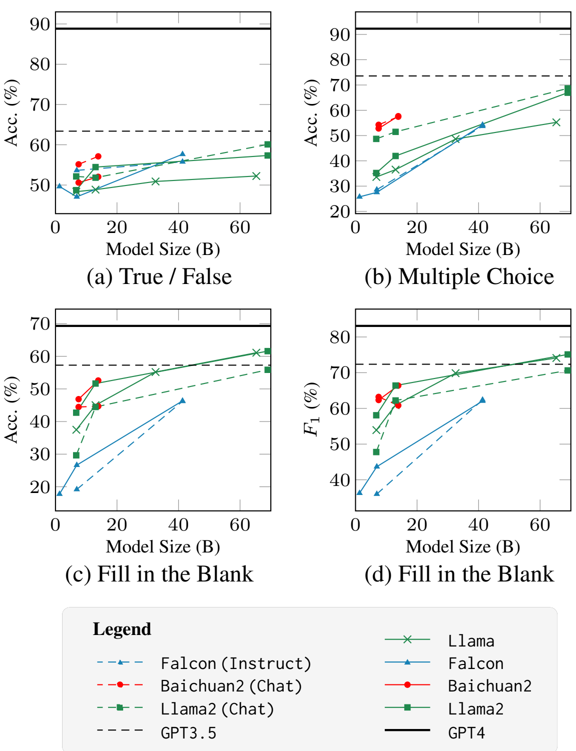


Figure 7: The performance of models with different sizes.

other question types. A typical example is that the Chat/Instruct versions have a higher accuracy on True/False and multiple choice questions than its foundation versions, but a lower accuracy on Fill in the Blank.

			Zero-shot				Few-shot			
			TF	MC	FITB		TF	MC	FITB	
			Acc.	Acc.	Acc.	F_1	Acc.	Acc.	Acc.	F_1
Kendall	TF	Acc.		0.560	0.480	0.516		0.674	0.490	0.471
	MC	Acc.	0.560		0.746	0.797	0.674		0.672	0.681
	FITB	Acc.	0.480	0.746		0.920	0.490	0.672		0.875
		F_1	0.516	0.797	0.920		0.471	0.681	0.875	
Pearson	TF	Acc.		0.798	0.698	0.671		0.798	0.698	0.671
	MC	Acc.	0.798		0.908	0.918	0.798		0.908	0.918
	FITB	Acc.	0.698	0.908		0.988	0.698	0.908		0.988
		F_1	0.671	0.918	0.988		0.671	0.918	0.988	

Table 7: The correlation coefficient between the metrics.

Models	GS	SC	DO	IO	MVP	ADJ	ADV	PPA	CO
Mistral 7B	58.75	53.49	54.71	60.79	50.11	47.73	43.97	30.87	56.69
Baichuan2 7B	53.98	44.82	49.56	68.57	52.49	46.23	43.77	26.83	49.31
Baichuan2 13B	57.62	53.99	57.75	63.03	54.13	45.90	47.52	26.93	56.70
Falcon 1B	28.85	28.48	29.90	37.50	27.61	25.90	27.50	26.42	32.77
Falcon 7B	35.76	33.67	33.81	52.65	35.58	32.14	27.77	24.85	40.20
Falcon 40B	53.06	43.30	48.41	62.22	51.72	48.00	42.17	33.25	40.94
Llama 7B	39.23	43.47	41.45	47.11	37.80	33.91	36.47	25.07	37.91
Llama 13B	46.42	42.86	48.42	53.45	39.05	44.13	36.71	29.96	39.62
Llama 30B	57.90	51.37	53.99	58.89	48.11	44.74	37.70	34.27	48.09
Llama 65B	59.73	57.59	62.67	65.48	55.51	55.72	47.31	44.69	62.92
Llama2 7B	47.99	45.29	43.57	52.32	44.71	33.50	35.44	26.49	36.15
Llama2 13B	55.20	43.77	50.15	63.44	50.00	37.22	35.81	29.19	45.87
Llama2 70B	62.02	57.90	64.53	72.64	57.67	60.29	61.28	42.91	62.85
Mistral 7B (Instruct)	59.46	49.23	61.97	57.12	57.44	48.09	47.89	32.24	52.00
Baichuan2 7B (Chat)	50.73	35.75	46.48	40.47	48.17	47.70	40.98	28.47	41.16
Baichuan2 13B (Chat)	59.22	51.42	56.07	56.29	49.48	46.14	48.24	32.81	58.45
Falcon 7B (Instruct)	37.23	36.31	36.82	38.53	28.40	27.81	26.08	25.93	40.27
Falcon 40B (Instruct)	57.71	46.20	50.68	61.52	52.76	44.98	44.36	29.49	50.59
Llama2 7B (Chat)	53.80	45.86	48.35	51.16	51.45	31.96	40.10	26.54	46.54
Llama2 13B (Chat)	51.17	48.75	52.80	59.13	49.94	42.99	40.10	30.32	53.41
Llama2 70B (Chat)	66.16	44.49	65.04	61.90	61.98	54.60	52.06	46.36	56.30
GPT3.5	72.84	66.69	64.82	68.66	62.00	66.41	55.18	42.40	55.41
GPT4	**87.08**	**86.74**	**82.25**	**88.33**	**81.93**	**89.58**	**66.70**	**75.44**	**74.47**
Avg.	53.41	47.43	51.48	57.10	49.00	45.26	42.22	33.21	48.59

Table 8: Overall performance of each model under **Zero**-shot setting at the knowledge point level.

Models	GS	SC	DO	IO	MVP	ADJ	ADV	PPA	CO
Mistral 7B	62.81	57.68	63.22	68.76	59.66	64.06	55.13	38.26	60.74
Baichuan2 7B	56.62	52.26	55.01	56.16	53.83	59.92	51.21	30.78	55.95
Baichuan2 13B	59.61	58.66	61.41	67.90	60.68	62.15	54.39	30.45	55.96
Falcon 1B	28.29	36.99	33.08	28.41	33.25	37.22	33.99	28.12	40.08
Falcon 7B	33.73	38.06	39.42	35.47	36.01	45.08	31.64	29.25	37.12
Falcon 40B	61.38	55.45	57.21	64.90	60.36	60.11	50.36	36.05	55.71
Llama 7B	40.42	47.52	47.38	45.25	40.48	51.53	40.38	26.45	43.71
Llama 13B	46.01	47.98	52.27	53.07	48.74	53.04	40.58	31.43	43.85
Llama 30B	60.55	52.63	55.79	72.41	56.16	60.04	53.88	34.72	52.21
Llama 65B	63.42	60.58	62.67	71.35	59.34	65.38	56.15	39.86	55.27
Llama2 7B	45.30	50.25	45.07	49.00	46.69	54.98	44.43	25.05	42.70
Llama2 13B	54.74	57.55	53.84	56.34	52.29	58.58	49.52	34.10	51.52
Llama2 70B	70.83	65.67	63.36	82.20	65.59	74.58	61.82	44.54	60.68
Mistral 7B (Instruct)	61.89	52.80	60.76	55.42	53.42	58.51	58.19	33.16	56.21
Baichuan2 7B (Chat)	58.26	50.67	56.51	51.37	55.13	62.64	54.37	37.64	54.74
Baichuan2 13B (Chat)	63.81	58.52	59.97	65.04	57.31	61.78	50.79	33.13	56.05
Falcon 7B (Instruct)	35.79	39.01	40.74	38.53	36.79	40.27	35.12	25.53	37.17
Falcon 40B (Instruct)	57.50	56.17	57.40	58.26	60.78	62.55	49.54	36.55	51.67
Llama2 7B (Chat)	51.69	48.54	52.23	47.51	51.86	46.86	43.13	27.96	46.09
Llama2 13B (Chat)	55.24	57.20	52.92	57.10	54.82	58.84	51.04	32.75	50.67
Llama2 70B (Chat)	68.86	56.22	67.14	68.76	70.36	75.04	60.00	49.72	56.33
GPT3.5	75.95	69.93	70.55	80.42	69.94	70.57	62.98	58.94	58.71
GPT4	**89.74**	**86.70**	**86.99**	**96.67**	**85.29**	**92.44**	**73.55**	**81.50**	**87.63**
Avg.	55.44	53.58	55.23	58.35	54.00	58.53	49.65	36.43	51.62

Table 9: Overall performance of each model under Few-shot setting at the knowledge point level.

Q. Types	**Models**	**GS**	**SC**	**DO**	**IO**	**MVP**	**ADJ**	**ADV**	**PPA**	**CO**
	`Mistral 7B`	56.15	56.15	53.33	40.00	54.09	49.73	40.00	50.91	55.76
	`Baichuan2 7B`	56.92	46.15	57.33	70.00	54.77	49.73	56.97	53.64	53.94
	`Baichuan2 13B`	61.54	56.92	59.33	43.33	58.18	48.65	48.48	40.00	51.52
	`Falcon 1B`	50.77	42.31	52.00	60.00	49.32	50.81	51.52	45.45	61.82
	`Falcon 7B`	50.77	49.23	50.67	46.67	50.00	42.16	41.21	49.09	71.52
	`Falcon 40B`	60.00	47.69	56.67	60.00	55.23	47.03	58.18	51.82	47.27
	`Llama 7B`	50.00	41.54	46.00	50.00	46.14	46.49	53.94	47.27	56.97
	`Llama 13B`	55.38	43.85	52.67	50.00	50.68	50.81	38.18	49.09	55.76
	`Llama 30B`	53.08	66.92	65.33	46.67	60.91	52.43	55.76	49.09	51.52
	`Llama 65B`	56.15	58.46	62.00	73.33	60.00	45.41	51.52	64.55	**72.73**
	`Llama2 7B`	60.00	47.69	56.67	60.00	55.23	50.27	60.00	51.82	47.27
TF (Acc.)	`Llama2 13B`	66.92	50.00	55.33	63.33	57.73	49.19	49.09	47.27	56.36
	`Llama2 70B`	62.31	53.08	62.67	66.67	60.00	54.59	60.61	51.82	52.12
	`Mistral 7B (Instruct)`	63.85	55.38	63.33	66.67	61.36	51.89	54.55	59.09	56.36
	`Baichuan2 7B (Chat)`	48.46	50.77	63.33	60.00	55.00	52.97	50.30	46.36	33.94
	`Baichuan2 13B (Chat)`	69.23	56.92	63.33	73.33	63.86	53.51	62.42	55.45	54.55
	`Falcon 7B (Instruct)`	49.23	46.92	52.67	56.67	50.23	45.95	41.21	50.00	**72.73**
	`Falcon 40B (Instruct)`	62.31	53.08	58.00	63.33	58.18	48.65	60.00	50.00	71.52
	`Llama2 7B (Chat)`	66.92	50.00	57.33	66.67	58.64	41.62	52.73	50.91	63.64
	`Llama2 13B (Chat)`	59.23	53.85	60.00	63.33	58.18	48.11	49.70	57.27	70.91
	`Llama2 70B (Chat)`	59.23	64.62	66.67	53.33	62.95	45.41	61.82	61.82	46.06
	`GPT3.5`	74.62	66.92	64.67	76.67	69.09	50.27	61.82	53.64	46.06
	`GPT4`	**90.77**	**92.31**	**85.33**	**80.00**	**88.64**	**82.70**	**75.15**	**88.18**	65.45
	`Mistral 7B`	59.05	47.06	58.62	65.00	48.82	53.33	60.00	22.00	46.25
	`Baichuan2 7B`	58.10	42.35	51.72	50.00	51.76	50.91	55.20	17.00	43.12
	`Baichuan2 13B`	55.24	50.59	67.59	75.00	61.76	51.52	60.00	25.00	54.37
	`Falcon 1B`	27.62	25.88	22.76	35.00	20.00	20.61	20.00	32.00	26.25
	`Falcon 7B`	26.67	21.18	21.38	40.00	27.65	33.33	25.60	19.00	24.38
	`Falcon 40B`	59.05	36.47	49.66	45.00	52.35	53.94	52.80	35.00	43.12
	`Llama 7B`	32.38	32.94	37.24	15.00	24.12	32.12	38.40	21.00	31.25
	`Llama 13B`	38.10	41.18	48.28	40.00	40.59	45.45	52.80	28.00	35.62
	`Llama 30B`	65.71	35.29	47.59	55.00	34.71	49.70	50.40	35.00	33.75
	`Llama 65B`	61.90	56.47	66.90	60.00	55.88	62.42	61.60	40.00	40.62
	`Llama2 7B`	44.76	36.47	35.86	25.00	31.76	33.94	36.80	22.00	33.75
MC (Acc.)	`Llama2 13B`	51.43	30.59	50.34	40.00	40.59	51.52	44.80	27.00	32.50
	`Llama2 70B`	80.95	60.00	75.17	80.00	63.53	70.91	70.40	45.00	57.50
	`Mistral 7B (Instruct)`	60.00	40.00	70.34	50.00	62.94	51.52	60.80	28.00	40.00
	`Baichuan2 7B (Chat)`	53.33	30.59	48.28	30.00	54.12	47.88	52.80	25.00	43.12
	`Baichuan2 13B (Chat)`	59.05	61.18	73.10	45.00	54.12	50.30	60.80	28.00	58.13
	`Falcon 7B (Instruct)`	29.52	38.82	31.03	25.00	25.88	25.45	28.00	19.00	26.88
	`Falcon 40B (Instruct)`	62.86	38.82	52.41	35.00	54.71	49.70	53.60	29.00	41.88
	`Llama2 7B (Chat)`	51.43	50.59	56.55	35.00	52.94	39.39	54.40	21.00	36.25
	`Llama2 13B (Chat)`	49.52	48.24	55.86	45.00	51.76	47.88	55.20	25.00	44.38
	`Llama2 70B (Chat)`	76.19	36.47	76.55	55.00	67.06	63.64	64.80	51.00	48.75
	`GPT3.5`	69.52	65.88	71.03	75.00	57.65	60.00	59.20	27.00	53.75
	`GPT4`	**91.43**	**91.76**	**88.97**	**95.00**	**90.59**	**94.55**	**78.40**	**86.00**	**82.50**

Table 10: Performance of each model under **Zero**-shot setting at the knowledge point level.

continued from previous page

Q. Types	Models	GS	SC	DO	IO	MVP	ADJ	ADV	PPA	CO
FITB (Acc.)	Mistral 7B	56.19	49.41	41.38	75.00	37.06	28.15	26.96	13.00	60.00
	Baichuan2 7B	41.90	36.47	28.97	85.00	41.76	26.67	12.17	6.00	41.94
	Baichuan2 13B	51.43	48.24	35.17	70.00	28.24	27.41	27.83	9.00	56.13
	Falcon 1B	3.81	9.41	6.21	15.00	8.24	1.48	6.09	0.00	3.23
	Falcon 7B	25.71	22.35	17.93	70.00	18.82	12.59	10.43	2.00	14.19
	Falcon 40B	32.38	40.00	28.97	80.00	37.06	33.33	10.43	8.00	18.71
	Llama 7B	30.48	48.24	30.34	75.00	34.12	12.59	11.30	3.00	14.19
	Llama 13B	40.00	36.47	33.10	70.00	18.82	25.93	13.91	9.00	18.71
	Llama 30B	50.48	45.88	39.31	75.00	35.88	23.70	2.61	14.00	48.39
	Llama 65B	57.14	48.24	50.34	60.00	37.65	49.63	21.74	23.00	67.74
	Llama2 7B	33.33	43.53	27.59	70.00	36.47	8.15	5.22	3.00	18.71
	Llama2 13B	41.90	43.53	34.48	85.00	41.76	5.19	9.57	8.00	40.00
	Llama2 70B	34.29	52.94	46.21	70.00	38.82	44.44	**45.22**	23.00	**72.90**
	Mistral 7B (Instruct)	49.52	42.35	41.38	50.00	37.06	31.11	22.61	5.00	50.32
	Baichuan2 7B (Chat)	45.71	17.65	16.55	25.00	25.29	29.63	16.52	5.00	32.90
	Baichuan2 13B (Chat)	43.81	25.88	19.31	40.00	17.06	24.44	17.39	7.00	52.26
	Falcon 7B (Instruct)	28.57	15.29	17.24	30.00	5.88	4.44	5.22	6.00	13.55
	Falcon 40B (Instruct)	41.90	40.00	31.03	85.00	34.71	25.19	13.91	7.00	29.03
	Llama2 7B (Chat)	37.14	27.06	19.31	45.00	31.76	6.67	7.83	6.00	27.74
	Llama2 13B (Chat)	38.10	34.12	30.34	65.00	28.24	20.00	10.43	5.00	32.90
	Llama2 70B (Chat)	59.05	23.53	38.62	75.00	43.53	44.44	24.35	18.00	66.45
	GPT3.5	70.48	60.00	48.97	45.00	47.65	84.44	40.00	36.00	56.77
	GPT4	**77.14**	**70.59**	**64.83**	**90.00**	**54.71**	**88.15**	42.61	**40.00**	67.74
FITB (F_1)	Mistral 7B	65.89	65.08	62.94	79.76	57.80	52.09	36.87	26.43	76.15
	Baichuan2 7B	51.96	55.45	50.30	86.43	60.13	49.43	26.12	13.72	59.78
	Baichuan2 13B	60.73	60.66	57.46	71.53	56.64	47.69	40.30	22.55	72.29
	Falcon 1B	12.52	25.10	23.69	20.02	18.78	11.06	15.88	3.64	17.25
	Falcon 7B	33.97	38.85	40.83	72.54	39.38	29.25	22.54	10.89	35.22
	Falcon 40B	47.88	51.47	48.87	83.33	58.11	52.75	20.61	17.84	46.14
	Llama 7B	40.16	63.60	51.87	77.68	52.17	33.68	22.81	10.89	36.84
	Llama 13B	51.55	50.66	55.55	70.70	32.96	46.33	24.41	16.57	36.24
	Llama 30B	59.33	57.89	58.81	75.00	61.54	40.46	11.26	23.45	69.61
	Llama 65B	65.14	67.44	67.90	66.21	63.67	69.04	35.91	36.04	83.07
	Llama2 7B	45.10	59.88	48.80	73.93	57.79	24.42	13.83	8.32	36.15
	Llama2 13B	52.58	57.89	55.05	89.00	61.61	16.70	17.52	18.58	57.50
	Llama2 70B	51.31	68.31	65.29	72.50	60.14	66.27	**60.46**	40.82	**84.93**
	Mistral 7B (Instruct)	59.57	62.23	63.10	59.38	58.95	50.62	34.04	14.26	68.94
	Baichuan2 7B (Chat)	55.06	34.13	39.11	37.85	45.48	54.84	23.13	23.12	59.95
	Baichuan2 13B (Chat)	54.97	46.45	44.24	61.08	43.88	44.78	25.62	22.94	73.11
	Falcon 7B (Instruct)	37.32	31.05	36.29	37.86	12.28	19.61	12.83	11.55	28.87
	Falcon 40B (Instruct)	54.05	53.41	52.21	87.43	56.06	48.00	25.05	11.93	47.74
	Llama2 7B (Chat)	48.97	46.90	43.00	58.62	53.80	23.04	18.53	9.41	51.70
	Llama2 13B (Chat)	51.43	54.21	54.72	73.10	51.53	45.95	20.38	12.36	56.97
	Llama2 70B (Chat)	67.04	41.24	65.19	79.72	68.31	65.06	34.78	34.51	81.76
	GPT3.5	78.30	74.54	68.57	63.65	70.87	93.46	49.06	57.14	76.05
	GPT4	**80.93**	**81.69**	**80.07**	**90.00**	**78.42**	**94.81**	50.48	**64.29**	83.19

Table 11: Performance of each model under **Zero**-shot setting at the knowledge point level (Continued).

Q. Types	Models	GS	SC	DO	IO	MVP	ADJ	ADV	PPA	CO
TF (Acc.)	Mistral 7B	55.38	53.59	57.78	52.22	55.45	54.23	53.13	57.58	64.44
	Baichuan2 7B	52.82	49.23	50.22	53.33	50.91	49.73	46.67	48.79	55.96
	Baichuan2 13B	52.82	50.77	52.89	62.22	52.88	50.81	51.52	48.48	54.14
	Falcon 1B	40.51	50.26	47.78	38.89	45.76	50.09	41.62	50.61	66.87
	Falcon 7B	47.18	51.03	46.44	46.67	48.03	46.49	44.44	48.79	46.67
	Falcon 40B	56.67	57.95	58.22	64.44	58.11	51.53	53.94	54.24	69.29
	Llama 7B	46.92	48.46	51.11	50.00	49.02	46.49	42.22	47.27	55.56
	Llama 13B	45.38	56.15	50.67	62.22	51.52	46.85	46.67	47.58	47.07
	Llama 30B	55.64	51.03	53.33	60.00	53.79	46.13	53.33	48.48	47.68
	Llama 65B	55.90	51.28	55.56	56.67	54.47	52.25	52.32	47.58	49.29
	Llama2 7B	45.64	52.82	50.67	57.78	50.30	46.85	46.46	46.97	50.10
	Llama2 13B	54.62	56.41	56.44	61.11	56.21	50.09	56.57	46.36	57.98
	Llama2 70B	61.79	57.95	54.44	71.11	58.79	58.74	59.39	56.97	50.10
	Mistral 7B (Instruct)	59.23	51.28	51.33	53.33	53.79	51.89	58.79	52.12	66.67
	Baichuan2 7B (Chat)	50.26	52.56	53.33	53.33	52.20	57.48	51.52	59.39	61.21
	Baichuan2 13B (Chat)	60.26	54.10	60.00	58.89	58.26	54.77	51.72	57.58	61.82
	Falcon 7B (Instruct)	53.33	51.79	55.78	55.56	53.86	49.19	53.74	45.15	63.64
	Falcon 40B (Instruct)	54.87	59.49	57.56	61.11	57.58	50.45	47.47	52.42	67.47
	Llama2 7B (Chat)	51.28	50.51	57.11	52.22	53.11	51.35	45.66	51.52	57.37
	Llama2 13B (Chat)	51.54	52.82	49.33	45.56	50.76	52.43	49.90	55.45	53.54
	Llama2 70B (Chat)	62.31	60.00	62.22	62.22	61.59	63.96	66.87	59.39	45.45
	GPT3.5	68.46	56.92	68.67	80.00	65.91	68.11	64.85	60.00	52.12
	GPT4	**87.69**	**93.08**	**92.67**	**90.00**	**91.14**	**89.19**	**83.64**	**86.36**	**89.09**
MC (Acc.)	Mistral 7B	66.98	58.43	71.95	68.33	62.16	53.54	63.47	24.00	45.62
	Baichuan2 7B	62.54	52.55	67.13	41.67	59.61	48.89	57.60	22.67	46.88
	Baichuan2 13B	64.13	60.00	77.47	61.67	69.22	55.76	58.40	21.00	45.83
	Falcon 1B	27.94	27.45	27.82	26.67	19.61	26.87	28.27	26.33	24.58
	Falcon 7B	29.84	30.98	29.66	15.00	25.10	28.28	26.67	27.00	26.67
	Falcon 40B	67.94	52.55	60.23	56.67	61.57	55.35	58.13	32.00	42.29
	Llama 7B	29.21	42.75	41.61	25.00	25.88	33.13	40.80	21.33	36.25
	Llama 13B	33.65	40.00	51.03	26.67	38.24	35.96	37.33	23.00	31.25
	Llama 30B	60.63	44.31	59.54	76.67	52.55	47.47	50.93	24.00	40.21
	Llama 65B	66.35	63.92	70.34	75.00	55.88	54.34	57.87	32.00	39.79
	Llama2 7B	39.05	38.04	34.71	11.67	37.65	40.00	44.53	15.67	31.88
	Llama2 13B	46.98	49.02	51.03	30.00	44.31	41.21	46.13	24.33	33.96
	Llama2 70B	80.00	69.41	76.32	88.33	68.63	75.76	65.33	35.67	55.83
	Mistral 7B (Instruct)	66.35	47.06	72.64	45.00	57.65	52.73	62.13	35.00	40.83
	Baichuan2 7B (Chat)	66.35	47.45	66.44	41.67	63.73	51.52	61.87	30.67	42.08
	Baichuan2 13B (Chat)	70.16	61.57	73.79	76.67	58.24	63.23	60.27	23.67	41.87
	Falcon 7B (Instruct)	27.30	33.33	29.43	21.67	30.78	30.10	28.80	19.33	28.75
	Falcon 40B (Instruct)	60.63	54.12	63.45	36.67	62.35	57.58	56.80	32.67	39.79
	Llama2 7B (Chat)	53.33	56.08	62.30	31.67	56.27	45.86	54.40	24.67	36.88
	Llama2 13B (Chat)	55.87	59.61	61.15	46.67	55.88	49.49	61.60	24.00	42.71
	Llama2 70B (Chat)	79.05	51.37	82.07	66.67	80.59	74.55	70.67	51.00	49.79
	GPT3.5	82.86	85.88	81.38	80.00	76.47	72.12	68.00	64.00	61.88
	GPT4	**95.24**	**92.94**	**94.48**	**100.00**	**91.76**	**95.15**	**81.60**	**95.00**	**91.25**

Table 12: Performance of each model under Few-shot setting at the knowledge point level.

continued from previous page

Q. Types	Models	GS	SC	DO	IO	MVP	ADJ	ADV	PPA	CO
FITB (Acc.)	`Mistral 7B`	62.22	53.33	49.20	85.00	53.33	79.26	40.87	28.67	64.73
	`Baichuan2 7B`	50.79	46.27	35.86	71.67	41.37	74.57	42.03	17.67	55.91
	`Baichuan2 13B`	58.41	56.86	44.14	76.67	51.37	72.84	46.09	17.33	62.15
	`Falcon 1B`	10.79	22.35	11.26	15.00	24.71	21.98	22.90	4.67	19.35
	`Falcon 7B`	19.05	21.57	31.03	43.33	26.27	50.12	15.94	9.00	27.53
	`Falcon 40B`	54.60	45.88	42.30	71.67	52.94	64.94	31.59	17.00	47.74
	`Llama 7B`	39.05	41.57	38.16	60.00	36.27	66.91	31.30	8.00	29.89
	`Llama 13B`	54.92	36.47	43.45	68.33	47.65	69.63	28.99	19.67	44.95
	`Llama 30B`	62.54	55.29	42.99	80.00	53.33	81.23	48.99	25.33	61.08
	`Llama 65B`	65.08	60.39	51.95	80.00	59.02	85.19	51.88	34.00	70.54
	`Llama2 7B`	47.30	51.37	38.62	75.00	43.53	70.62	34.20	9.33	37.20
	`Llama2 13B`	58.41	60.00	43.68	76.67	47.06	79.26	38.84	25.67	53.98
	`Llama2 70B`	66.98	63.53	49.43	86.67	60.00	85.43	**53.91**	34.00	69.46
	`Mistral 7B (Instruct)`	56.19	51.76	46.67	63.33	37.84	62.96	47.54	7.33	51.18
	`Baichuan2 7B (Chat)`	51.75	43.14	36.09	53.33	36.27	71.85	42.32	16.33	51.61
	`Baichuan2 13B (Chat)`	56.19	52.16	34.25	56.67	45.69	58.27	32.75	13.00	57.42
	`Falcon 7B (Instruct)`	21.59	20.00	25.75	33.33	18.04	29.38	15.36	8.67	11.18
	`Falcon 40B (Instruct)`	52.70	45.10	39.77	75.00	52.94	72.59	36.81	19.33	38.71
	`Llama2 7B (Chat)`	46.35	29.41	25.52	51.67	37.45	31.60	22.03	5.67	30.32
	`Llama2 13B (Chat)`	53.33	49.02	35.86	76.67	46.47	66.42	35.36	15.33	44.09
	`Llama2 70B (Chat)`	60.95	47.84	46.21	76.67	59.22	82.22	37.10	31.00	66.45
	`GPT3.5`	73.33	60.00	50.34	80.00	55.88	68.15	48.70	42.00	56.77
	`GPT4`	**85.71**	**67.06**	**64.83**	**100.00**	**61.76**	**89.63**	46.96	**54.00**	**76.77**
FITB (F_1)	`Mistral 7B`	69.88	68.68	70.67	86.43	69.39	89.55	56.73	37.77	79.56
	`Baichuan2 7B`	58.21	63.73	59.48	75.32	60.55	87.72	56.68	24.11	74.12
	`Baichuan2 13B`	65.34	73.57	63.59	82.97	68.51	86.94	60.42	26.39	73.68
	`Falcon 1B`	22.08	44.16	36.04	24.34	44.04	47.40	41.26	10.19	38.22
	`Falcon 7B`	29.30	42.78	53.30	46.13	43.54	70.83	31.70	14.95	48.50
	`Falcon 40B`	64.48	65.80	64.04	75.53	69.86	81.92	46.41	26.82	63.32
	`Llama 7B`	51.22	61.15	60.71	61.47	56.79	83.01	44.91	13.51	48.76
	`Llama 13B`	63.06	59.10	66.75	72.32	65.32	83.01	46.47	27.75	61.49
	`Llama 30B`	68.21	69.83	66.01	81.11	70.97	91.82	65.75	38.03	76.42
	`Llama 65B`	70.96	72.70	72.25	84.76	76.30	93.89	64.65	45.98	82.93
	`Llama2 7B`	55.10	68.43	61.03	80.12	60.71	85.55	50.38	15.70	55.05
	`Llama2 13B`	66.83	74.46	64.38	79.15	65.60	89.59	52.89	37.56	71.24
	`Llama2 70B`	74.40	75.79	69.19	87.62	78.69	93.04	**67.53**	47.96	82.72
	`Mistral 7B (Instruct)`	63.98	68.35	69.96	72.52	59.81	78.84	59.73	17.40	71.09
	`Baichuan2 7B (Chat)`	64.62	60.88	63.41	64.92	62.66	86.02	57.14	29.36	70.24
	`Baichuan2 13B (Chat)`	65.83	67.63	58.01	62.46	65.21	76.41	48.05	23.29	71.48
	`Falcon 7B (Instruct)`	31.87	43.82	48.30	43.40	33.39	53.66	30.28	15.52	27.05
	`Falcon 40B (Instruct)`	61.28	64.69	62.65	79.03	71.88	86.67	51.90	29.79	56.75
	`Llama2 7B (Chat)`	54.53	48.64	49.07	65.60	54.98	55.16	36.66	9.73	57.70
	`Llama2 13B (Chat)`	63.30	69.30	60.70	81.48	69.20	82.75	47.87	22.27	67.45
	`Llama2 70B (Chat)`	69.48	66.73	68.07	78.10	78.56	90.97	47.84	46.54	81.03
	`GPT3.5`	79.74	73.95	72.89	82.50	78.98	74.84	63.50	63.67	67.47
	`GPT4`	**86.88**	**81.12**	**82.83**	**100.00**	**84.16**	**96.30**	63.84	**72.29**	**88.30**

Table 13: Performance of each model under Few-shot setting at the knowledge point level (Continued).

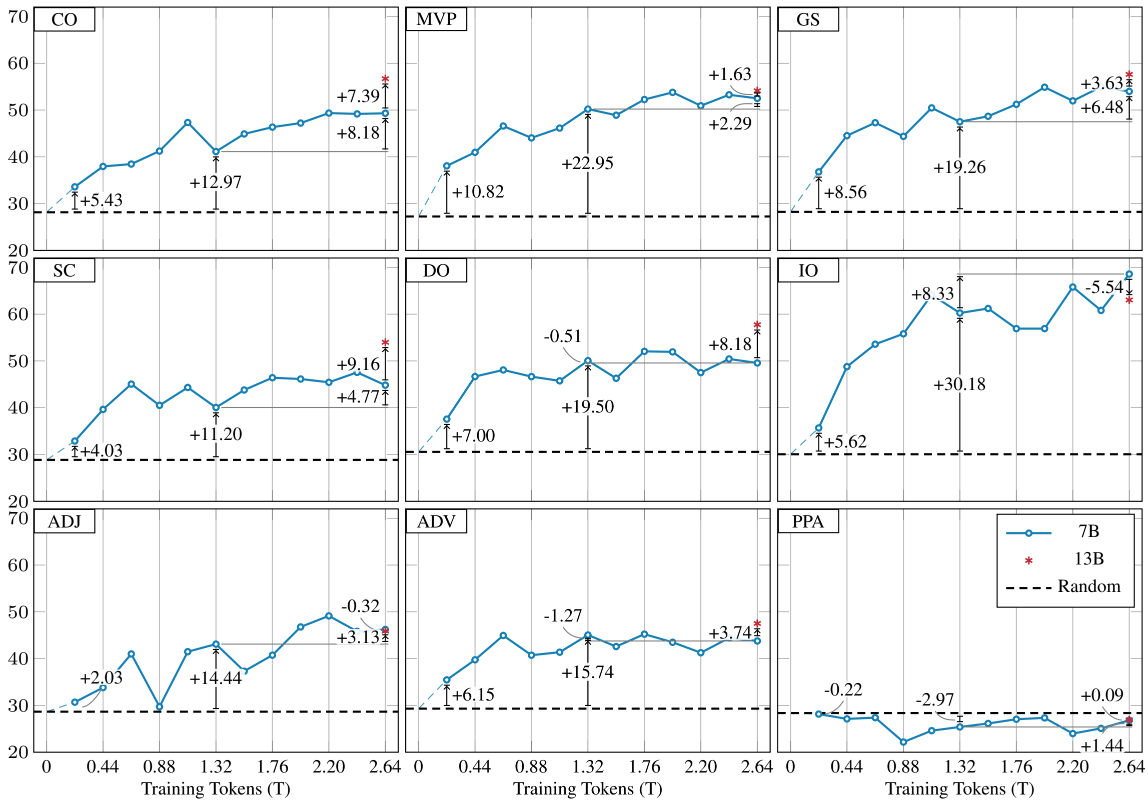


Figure 8: The overall scores of BaiChuan-2 intermediate checkpoints under Zero-shot setting with different numbers of training tokens.